

1

IREM *

TITLE GT40 ROM BOOTSTRAP TEST MAINDEC-11-DDGTD-B

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DDGTD-B-D
PRODUCT NAME: GT40 ROM VERIFY
DATE CREATED: NOVEMBER 1, 1973
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: RAYMOND SHOOP

COPYRIGHT (C) 1973, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED TO PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DEC'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DEC.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.

DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.

1. ABSTRACT

THE DDGTD-B DIAGNOSTIC PROGRAM IS WRITTEN TO BE USED AS AN AID
TO HARDWARE DEBUGGING AND MAINTENANCE OF THE GT40 ROM
BOOTSTRAP LOADER VERSION 1 OR 2.

THE AVAILABLE TESTS ARE
PRG0 = LOGIC TESTS
PRG1 = ROM DATA DUMP TO THE CONSOLE TELETYPE
PRG2 = SINGLE ROM ADDRESS READ DATA LOOP

2. REQUIREMENTS

2.1 EQUIPMENT

GT40 DISPLAY PROCESSOR WITH ROM BOOTSTRAP VERSION 1 OR 2.

2.2 STORAGE

THIS PROGRAM USES MEMORY LOCATIONS 0-7776 + 16000-16776(8).

3. LOADING PROCEDURE

PROCEDURE FOR A NORMAL BINARY TAPE SHOULD BE FOLLOWED.

4. STARTING PROCEDURE

4.1 STARTING ADDRESSES

0200 PROGRAM 0, ROM LOGIC TEST,
0204 PROGRAM 1, ROM DATA DUMP ON CONSOLE TTY,
0210 PROGRAM 2, SINGLE ROM READ,

4.2 SWITCH SETTINGS

CONSOLE SW 11=0	NORMAL RUN (64, INTERACTIONS/TEST)
CONSOLE SW 11=1	SUPPRES SUBPROGRAM INTERACTIONS
CONSOLE SW 08=0	TEST AS VERSION 2 ROM (512, WORDS)
CONSOLE SW 08=1	TEST AS VERSION 1 ROM (256, WORDS)

5" PROGRAM DESCRIPTIONS

5.1 PRG0 - LOGIC TESTS

THE LOGIC TESTS CONSIST OF 4 ROUTINES TO TEST THE GT40 ROM
BOOTSTRAP LOGIC

5.1.1 ROUTINE DESCRIPTIONS

ROUTINE	TESTS
T1	ADDRESSABILITY OF GT40 ROM BOOTSTRAP
T2	DATA RELIABILITY
T3	THAT GT40 ROM BOOTSTRAP TIMES OUT WHEN REFERENCED BY A DATA BUS CYCLE
T4	THAT DATA READ FROM THE ROM IS CORRECT

5.2 PRG1 - ROM DATA DUMP

THIS PROGRAM TYPES OUT THE 512,720, WORDS OF ROM DATA ON THE
CONSOLE TELETYPE AND HALTS.

5.3 PRG2 - SINGLE ROM ADDRESS READ DATA LOOP

THIS PROGRAM CONTINUOUSLY READS DATA FROM THE ADDRESS IN THE
CONSOLE SWITCH REGISTER.

6" ERRORS

THE PROGRAM WILL ONLY HALT ON ERROR, THE PROGRAM DOES NOT
CONTAIN FACILITIES FOR REPORTING ERROR CONDITIONS.
TO PLACE THE PROGRAM INTO A SCOPE LOOP, REPLACE THE ERROR
HALT WITH A NOP.

7" EXECUTION TIME

PRG0 TAKES APPROX. 5 SECONDS PER PASS.
PRG1 N/A
PRG2 N/A

```

151
152
153
154
155
156
157
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179

```

.LIST ME,BIN,SEQ,LD
 .ENABL ABS,AMA
 ;LOAD ADDRESS=0200
 ;DEPRESS START
 ;STACK POINTER IS AT 900
 .LIST
 .B34
 SCOPEC
 0
 ;EQUATE STATEMENTS
 SCOPESTRAP
 TPCSR=177564
 TPDOR=177564
 PSW=177776
 SR=177570
 DISLAY=177570
 STKPTR=900
 ;200
 JMP PRNTRS
 JMP PRO
 JMP PRO

;ADDRESS OF DISPLAY REGISTER
 ;INITIAL STACK SETTING

```

181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222

```

;1000
 ROMADD 166000
 WORDS 512
 IMAGE START 12500
 OSR 172002
 FILLER 10
 FILLCNT 10
 ICNT 0
 DUMP 0
 CHARA 0
 TERM 0
 PRNTRS MOV STKPTR,900
 JSR PC,INITON
 ;SET STACK PTR
 ;CHECK ROM VERSION
 ;PROGRAM LOGIC TESTS
 PRG0 CLR ICNT
 PRG0 MOV STKPTR,900
 MOV PRG0,RETURN
 MOV ICNT,0
 ;CLEAR PASS COUNT
 ;SET RETURN ADDRESS FOR SCOPE
 ;DISPLAY PASS COUNT
 ;TEST1 TEST ABILITY TO REFERENCE ROM WITHOUT TIMING OUT
 ;11 MOV ROMADD,90
 ;11 MOV WORDS,512
 ;11 MOV ERROR,4
 ;11 MOV (0),93
 ;11 TST (0)
 ;11 ADD -(0),DUMP
 ;11 CMP (0),(0)
 ;11 RTS
 ;11 SUB -(0),DUMP
 ;11 ADD #2,0
 ;11 DEC X1
 ;11 BNE T1A
 ;11 BR T1B
 ;11 CMP (0),(0)
 ;11 HALT
 ;11 BR T1A
 ;11 SCOPE
 ;GET ROM ADDRESS
 ;GET ADDRESS COUNTER
 ;SET UP TIME OUT VECTOR
 ;REFERENCE
 ;ROM
 ;INCREMENT POINTER
 ;DECREMENT ADDRESS COUNTER
 ;BRANCH IF NOT FINISHED
 ;GO TO SCOPE LOOP
 ;REPOSITION STACK
 ;ERROR, TIME-OUT ON ROM ADDRESS
 ;LOOP ON ERROR

```

224                                     TEST THAT RDN DATA CAN BE READ RELIABLY.
225
226
227 001140 013700 001000      TEST      MOV      R0HADD,%0      GET RDN ADDRESS
228 001144 013701 001002      MOV      R0HADD,%1      GET ADDRESS COUNTER
229 001150 012737 000000 000004  T3A(      MOV      %0,%1      INITIALISE TIME OUT VECTOR
230 001156 000037 001010      DUMP      DUMP      INITIALISE DUMP
231 001162 011003 001010      MOV      (%0),%1      GET DATA
232 001164 002037 001010      ADD      (%0),%1      ADD DATA TO DUMP
233 001170 103703 001010      SUB      DUMP,%1      SUBTRACT DATA FROM DATA
234 001174 001402 001010      BEQ      T3B      BRANCH IF EQUAL
235 001176 000000 001010      ERRCR21 HALT      DATA ERROR
236 001200 000764 001010      BR      T3A      LOOP ON ERROR
237 001202 044037 001010      T3B(      BIC      =(%0),%1      CLEAR DUMP BIT
238 001206 001402 001010      BEQ      T3C      BRANCH IF EQUAL TO 0
239 001210 000000 001010      HALT      DATA ERROR
240 001212 000773 001010      BR      T3B      LOOP ON ERROR
241 001214 021010 001010      T3C(      CMP      (%0),(%0)      COMPARE DATA
242 001216 001402 001010      BEQ      T3D      BRANCH IF EQUAL
243 001220 000000 001010      HALT      DATA ERROR
244 001222 000774 001010      BR      T3C      LOOP ON ERROR
245 001224 122040 001010      T3D(      CMPS      (%0)+(%0)      COMPARE DATA (BYTE OPERATION)
246 001226 001402 001010      BEQ      T3E      BRANCH IF EQUAL
247 001230 000000 001010      HALT      DATA ERROR
248 001232 000774 001010      BR      T3D      LOOP ON ERROR
249 001234 000720 001010      T3E(      TEST      (%0)      INCREMENT ADDRESS POINTER
250 001236 000301 001010      DEC      %1      DECREMENT ADDRESS COUNTER
251 001240 001344 001010      BNE      T3A      RETURN IF NOT DONE
252 001242 104400 001010      SCOPE
253

```

```

255                                     TESTS TEST THAT RDN TIMES OUT IF REFERENCED BY OTHER
256                                     THAN DATA SUB CYCLE
257
258
259 001244 012704 000500      T3T      MOV      %0HPTR,%0      GET STACK PTR
260 001250 013700 001000      MOV      R0HADD,%0      GET RDN ADDRESS
261 001254 013701 001002      MOV      R0HADD,%1      GET ADDRESS COUNTER
262 001260 012737 001274 000004  T3AA(      MOV      %T3B,%4      GET UP TIME OUT VECTOR
263 001266 010010 001274 000004  T3AI      MOV      %0,%1      ATTEMPT TO ALTER DATA
264 001270 000000 001274 000004  T3B(      HALT      HERE IF DID NOT TIME OUT
265 001272 000775 001274 000004  T3B(      BR      T3A      LOOP ON ERROR
266 001274 012737 001312 000004  T3B(      MOV      %T3B,%4      GET UP TIME OUT VECTOR
267 001276 022626 001312 000004  T3C(      CMP      (%0)+(%0)      REPOSITION STACK
268 001278 0003210 001312 000004  T3C(      INC      (%0)      ATTEMPT TO ALTER DATA
269 001280 000000 001312 000004  T3C(      HALT      HERE IF DID NOT TIME OUT
270 001282 000775 001312 000004  T3D(      BR      T3D      LOOP ON ERROR
271 001284 012737 001332 000004  T3D(      MOV      %T3B,%4      GET UP TIME OUT VECTOR
272 001286 022626 001332 000004  T3E(      CMP      (%0)+(%0)      REPOSITION STACK
273 001288 000577 177492 001332 000004  T3E(      CLR      %R0HADD      ATTEMPT TO ALTER DATA
274 001290 000000 001332 000004  T3F(      HALT      HERE IF DID NOT TIME OUT
275 001292 000774 001332 000004  T3F(      BR      T3E      LOOP ON ERROR
276 001294 000320 001332 000004  T3F(      TEST      (%0)      INCREMENT ADDRESS POINTER
277 001296 022626 001332 000004  T3F(      DMP      (%0)+(%0)      REPOSITION STACK
278 001298 000301 001332 000004  T3F(      DEC      %1      DECREMENT ADDRESS COUNTER
279 001300 001347 001332 000004  T3F(      BNE      T3AA      RETURN IF NOT DONE
280 001302 012737 001332 000004  T3F(      MOV      %6,%4      RESTORE TIME OUT THAP
281 001304 104400 001332 000004  T3F(      SCOPE      SCOPE LOOP

```

```

283                                     ;COMPARE THE ROM DATA TO THE IMAGE DATA
284                                     ;
285                                     ;R0=WORD NUMBER
286                                     ;R1=GOOD ADDRESS
287                                     ;R2=GOOD DATA
288                                     ;R3=BAD ADDRESS
289                                     ;R4=BAD DATA
290
291 001392 012708 000000          T4T  MOV  #0,00          ;SET UP INITIAL WORD COUNT
292 001396 013701 001004          MOV  IMAGE,X1          ;SET UP STARTING ADDRESS OF ROM IMAGE
293 001342 013703 001000          MOV  ROMADD,X3          ;SET UP STARTING ROM ADDRESS
294 001346 011102                  T4A:  MOV  (X1),X2          ;READ EXPECTED VALUE
295 001370 011304                  MOV  (X3),X4          ;READ ROM VALUE
296 001372 020204                  CMP  X2,X4          ;COMPARE EXPECTED TO THE VALUE READ
297 001374 001402                  BEQ  T4B          ;BRANCH IF CORRECT
298 001376 000000                  HALT          ;LEARN: ROM VALUE FAILED TO EQUAL EXPECTED
299 001400 000772                  BR   T4A
300
301 001402 022123                  T4B:  CMP  (X1+,(X3)),X0          ;INCREMENT ADDRESSES POINTERS
302 001404 005200                  INC  X0          ;INCREMENT WORD COUNT
303 001406 023700 001002          CMP  WORDS,X0          ;COMPARE IF END WORD
304 001412 001305                  BNE  T4A          ;BRANCH IF NOT LAST WORD
305 001414 104400                  T4E:  SCDE
306
307 001416 005237 001014          END:  INC  COUNT          ;INCREMENT PASS COUNT
308 001422 012777 000001 177356  MOV  #1,00SR          ;RING THE GY40 BELL
309 001430 012737 000207 177566  DDNE0:  MOV  #27,00TPDR          ;RING THE TELETYPE BELL
310 001436 105737 177564          MOV  TSTB          ;
311 001442 100375                  SPL          ;
312 001444 012737 000207 177566  MOV  #27,00TPDR          ;
313 001452 105737 177564          TSTB          ;
314 001456 100375                  SPL          ;
315 001460 013700 000042          MOV  #40,00          ;RETURN TO DECTAPE MONITOR
316 001464 001406                  BEQ  000011
317 001466 000005                  RESET
318 001470 000005                  RESET
319 001472 004710                  JSR  7,00          ;RETURN
320 001474 000240                  NOP
321 001476 000240                  NOP
322 001500 000240                  NOP
323 001502 000137 001034          DONE1:  JMP  PROG
324

```

```

326                                     ;THIS PROGRAM TYPES OUT ROM DATA
327                                     ;
328                                     ;
329 001506 012706 000500          PRG1:  MOV  #SYNTRX6          ;INITIALIZE STACK
330 001512 012737 000006 000004  MOV  #0,00          ;SET UP BUSS ERROR
331 001520 004737 002306          JSR  PC,00SWTCH          ;
332 001524 004537 001712          JSR  5,TYPEM          ;
333 001530 002275                  M0
334 001532 004537 001712          JSR  5,TYPEM          ;TYPE MESSAGE
335 001536 002275                  M1          ;FROM DATA
336 001540 013701 001002          MOV  WORDS,X1          ;GET # OF WORDS
337 001544 013700 001000          PRG1A:  MOV  ROMADD,X0          ;GET STARTING ADDRESS
338 001550 012702 000010          MOV  #10,X2          ;GET ADDRESS INDICATOR
339 001554 105737 177564          TSTB          ;
340 001560 100375                  SPL          ;
341 001562 100037 002144          PRG1B:  MOV  #0,00STYP          ;TELETYPE FLAG
342 001566 004737 002146          JSR  7,00A          ;GET ADDRESS
343 001572 004537 001712          JSR  5,TYPEM          ;AND TYPE IT
344 001576 002301                  M0          ;
345 001600 012037 002144          PRG1C:  MOV  (0)+,00STYP          ;OR/LP
346 001604 004737 002146          JSR  7,00A          ;TYPE
347 001610 105737 177564          TSTB          ;DATA
348 001614 100375                  SPL          ;
349 001616 012737 000040 177566  MOV  #1,7PDR          ;WAIT FOR
350 001624 005301                  DEC  PROG          ;TELETYPE FLAG
351 001626 001410                  BEQ  PROG          ;TYPE SPACE
352 001630 005302                  BEQ  PROG          ;ALL DATA TYPED
353 001632 001302                  BNE  PROG          ;GO TO FINISH
354 001634 012702 000010          MOV  #10,X2          ;RETURN TO PRG1C
355 001640 004537 001712          JSR  5,TYPEM          ;GET ADDRESS INDICATOR
356 001644 002275                  M0          ;
357 001646 000745                  BR   PRG1B          ;
358 001650 004537 001712          PRG1D:  JSR  5,TYPEM          ;
359 001654 002275                  M0          ;
360 001656 004537 001712          JSR  5,TYPEM          ;
361 001662 002275                  M0          ;
362 001664 000000                  HALT          ;
363 001666 000707                  BR   PRG1
364
365                                     ;ROUTINE TO LOAD ON A SINGLE ADDRESS
366                                     ;
367 001670 012706 000500          PRG2:  MOV  #SYNTRX6          ;
368 001674 012737 000006 000004  MOV  #0,00          ;
369 001702 013700 177570          MOV  #0,00          ;
370 001706 011001                  MOV  (0),X1          ;
371 001710 000767                  BR   PROG

```

ROUTINE TO TYPE A MESSAGE

```
373
374
375 001712 010026 TYPEM1 MOV R0,(6)* ;SAVE REGISTER 0
376 001714 012500 MOV (5)*R0 ;PLACE MESSAGE ADDRESS IN R0
377 001716 112037 001022 MOV0 (0)*TERM ;GET TERMINATOR CHARACTER
378 001722 112037 001020 TYPEM1 MOV0 (0)*CHARA ;GET NEXT CHARACTER
379 001726 123737 001020 001022 CHAPA,TERM ;WAS NEXT CHARACTER THE TERM
380 001734 001000 BNE TYPEM0 ;CHARACTER
381 001736 014000 MOV =16)*R0 ;RESTORE R0
382 001740 105737 177564 TST0 TPOBR
383 001744 100378 BPL =4
384 001746 000205 RTS ;AND EXIT
385 001750 123727 001020 000045 TYPEM0 CHAPA,01X ;WAS CHARACTER X
386 001756 001027 BNE TYPEM0 ;TEST TELEPRINTER FLAG
387 001760 105737 177564 TST0 TPOBR ;AND WAIT FOR DONE
388 001764 100378 BPL =4 ;LOAD TELEPRINTER WITH PAR, RET
389 001766 012737 000210 177566 MOV #21)*TPOBR ;LOAD FILLER COUNT
390 001774 013757 001010 001012 MOV FILLER,FILCNT ;LOAD FILLER COUNT
391 001002 000403 BR 11
392 001004 012737 000000 177566 2ST MOV #0)*TPOBR ;PRINT FILLER CHAR
393 001006 105737 177564 1ST TST0 TPOBR ;TEST TELEPRINTER FLAG
394 001008 100378 BPL =4 ;AND WAIT FOR DONE
395 001010 005337 001012 OLC FILCNT ;FINISHED FILLERS
396 001014 001364 BNE =5 ;IF NOT
397 001016 012737 000212 177566 MOV #21)*TPOBR ;LOAD TELEPRINTER WITH LINE FEED
398 001018 000732 BR TYPEM1 ;GET NEXT CHARACTER
399 001020 105737 177564 TYPEM0 TST0 TPOBR ;TEST TELEPRINTER FLAG
400 001022 100378 BPL =4 ;AND WAIT FOR DONE
401 001024 013737 001020 177566 MOV CHAPA,TPOBR ;LOAD TELEPRINTER BUFFER
402 001026 000723 BR TYPEM1 ;AND GET NEXT CHARACTER
403
404
405
406 001004 032737 040000 177570 SCOPE0 BIT #40000:0R ;TEST BR FOR SCOPE
407 001006 001023 BNE SCOPE0 ;YES SCOPE
408 001008 032737 004000 177570 BIT #40000:0R ;TEST FOR ITERATION
409 001010 001007 BNE SCOPE0 ;INHIBIT ITERATION
410 001012 023737 002126 002124 CMP #SCOPE0,ICOUNT ;ITERATION COMPLETE
411 001014 001403 BEO SCOPE0 ;ITERATION COMPLETE GO TO SCOPE0
412 001016 005237 002126 INC SCOPE0 ;INCREMENT ITERATION COUNT
413 001018 000410 BR SCOPE0 ;GO TO SCOPE0
414 001020 005037 002126 SCOPE0 CLR SCOPE0 ;CLEAR ITERATION COUNT
415 001022 011637 002130 MOV #0)*RETURN ;GET ADDRESS OF NEXT TEST
416 001024 000002 RTI ;EXIT
417 001026 000100 ICOUNT0 100 ;CONTAINS SUBTEST ITERATION COUNT
418 001028 000000 SCOPE0 0 ;CONTAINS RETURN PC FOR SCOPE
419 001030 000000 RETURN0 WORD 0 ;POP PC
420 001032 005726 177776 2ST TST (6)* ;TEST SCOPE CONDITION CODES
421 001034 012637 177776 MOV (6)*TPOBR
422 001036 000177 177764 JMP #RETURN
423
```

THIS ROUTINE CONVERTS AN OCTAL NUMBER TO ASCII AND TYPES IT ON THE TTY.

```
425
426
427
428
429 001144 000000 020TYP0 0
430 001146 013746 177564 02A1 MOV TPOBR,=(6) ;SAVE TPOBR
431 001148 010246 MOV R2,=(6) ;SAVE R2
432 001150 010146 MOV R1,=(6) ;SAVE R1
433 001152 010046 MOV R0,=(6) ;SAVE R0
434 001154 013700 002144 MOV 020TYP0,R0 ;GET DATA TO BE TYPED
435 001156 012700 000000 MOV #0)*R1 ;GET COUNTER
436 001158 005002 OLP R2 ;CLEAR WORKING REGISTER
437 001160 006100 ROL R0 ;MOV FIRST BIT (MSB) INTO
438 001162 006102 ROL R2 ;R2
439 001164 006202 000260 02AA1 ADD #20)*R2 ;FORM ASCII CODE
440 001166 105737 177564 TST0 TPOBR ;TEST TELEPRINTER
441 001168 100378 BPL =4 ;FLAG AND WAIT UNTIL DONE
442 001170 010007 177566 MOV R2,TPOBR ;LOAD TELEPRINTER BUFFER
443 001172 005002 OLP R2 ;CLEAR WORKING REGISTER
444 001174 006100 ROL R0 ;ROTATE THE
445 001176 006102 ROL R2 ;NEXT
446 001178 006100 ROL R0 ;OCTAL CHARACTER
447 001180 006102 ROL R2 ;INTO
448 001182 006100 ROL R0 ;REGISTER
449 001184 006102 ROL R2 ;TWO
450 001186 005301 DEC R1 ;DECREMENT COUNTER
451 001188 001300 BNE 02AA ;GO TO 02AA IF NOT 0
452 001190 012600 MOV (0)*R0 ;FINISHED, RESTORE REGISTERS
453 001192 012601 MOV (0)*R1
454 001194 012602 MOV (0)*R2
455 001196 012607 177564 MOV (0)*TPOBR ;AND TPOBR
456 001198 000207 RTS ;AND EXIT
457
458
459 001202 002500 000107 032055 IASCII MESSAGES ;MSG=48 ROM DATAX00
460 001204 002500 047522 000115 M0T ASCII ;
461 001206 040554 022448 000115 M0T ASCII ;
462 001208 100 022448 100 M0T ASCII ;
463 001210 100 022448 100 M0T ASCII ;
464 001212 032737 000400 177570 SWITCH0 BIT #40000:0R ;TEST BIT 0
465 001214 001007 BNE 11 ;IF BIT 0
466 001216 012737 001000 001002 MOV #0)*WORDS ;SET UP VERSION 2 LENGTH
467 001218 012737 000000 001004 MOV #0)*IMAGE ;SET UP VERSION 2 STARTING ADDR
468 001220 000406 BR 25 ;
469 001222 012737 000400 001002 1ST MOV #0)*WORDS ;SET UP VERSION 1 LENGTH
470 001224 012737 010000 001004 MOV #0)*IMAGE ;SET UP VERSION 1 STARTING ADDR
471 001226 000207 2ST RTS ;
472
473 ;SETTL ROM VERSION 2 VALUES
```

475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528

```

*****
;
;   EXCEPT FOR THE NEW ORIGIN ADDRESS AND SEVERAL "1148288"
;   FOR ADDRESS PADDING THIS IS AN EXACT COPY OF THE CONTENTS
;   OF THE GT-48 BOOTSTRAP VERSION #2
;
*****

```

.TITLE SCROLLING ROM BOOTSTRAP FOR THE GT48

; BOOTSTRAP.T16 OCT 18, 1973

```

;
;   COPYRIGHT 1973, DIGITAL EQUIPMENT CORPORATION
;   140 MAIN STREET
;   MAYNARD, MASSACHUSETTS 01754
;

```

; WRITTEN BY JACK BURNES.

```

;
;   THIS PROGRAM IS THE SECOND VERSION THE THE ROM BOOTSTRAP FOR
;   THE GT48 DISPLAY TERMINAL. IT INCLUDES SCROLLING AND AN END OF
;   MEMORY SEARCH FOR THE LOADER.
;

```

.ENABL ABS,AMA ;ASSEMBLER DIRECTIVES FOR ABSOLUTE BINARY OUTPUT
 ; NOTE! USE "MACOLX" TO ASSEMBLE THIS PROGRAM,

.SECTL DEFINITION SECTION
 .PAGE

529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582

REGISTER DEFINITIONS

BASIC DEFINITIONS

!DEFINE STANDARD VALUES.

```

000000 R00X0
000001 R10X1
000002 R20X2
000003 R30X3
000004 R40X4
000005 R50X5
000006 SP0X6
000007 PC0X7

```

GT48 DEFINITIONS

```

000000 CHAR0R0
000001 POINT0R1
000002 TABCNT0R2
000003 SCAN0R3
000004 HOLD0R4
000005 COUNT0R5

```

!CONTAINS THE INPUT CHARACTER.
 !POINTS TO NEXT INSERTION BYTE IN DISPLAY BUFFER
 !CHARACTER COUNTER FOR THE "TAB" FEATURE.
 !GENERALLY CONTAINS A POINTER WHICH
 !IS USED WHEN SCANNING MEMORY FOR SOMETHING.
 !TYPICALLY A TEMPORARY WHICH IS USED TO RETAIN
 !A VALUE FOR A SHORT TIME.
 !TYPICALLY USED AS A COUNTER.

LOADER DEFINITIONS

```

000000 L.BYTE=CHAR
000001 L.ADR=POINTR
000002 L.BC=SCAN0R
000003 L.CKSUM=COUNTR
000004 INDEX=SCAN

```

!CHARACTER INPUT FOR THE LOADER.
 !CURRENT MEMORY ADDRESS TO BE LOADED.
 !NUMBER OF DATA ITEMS TO LOAD.
 !CHECKSUM ON THE INPUT DATA.
 !INDICATES HOW TO ASSEMBLE THE 8 BIT CHARACTER.

.PAGE

```

583
584
585
586
587
588
589
590
166800      ORIGIN#166800      !ORIGIN OF THE BOOTSTRAP.
591
592      175610      OL11#175610      !INPUT STATUS REGISTER OF OL11
593      175612      OL11#OL11#2      !INPUT CHARACTER FROM OL11
594      175614      OL11#OL11#2      !OUTPUT STATUS OF THE OL11
595      175616      OL11#OL11#2      !OUTPUT CHARACTER TO THE OL11
596
597      177560      KBDIS#177560      !KEYBOARD INPUT STATUS
598      177562      KBDIS#KBDIS#2      !CURRENT CHARACTER FROM KEYBOARD.
599
600      172000      GT48#172000      !GT48 PROGRAM COUNTER.
601      172002      GT48#GT48#2      !GT48 STATUS REGISTER ADDRESS.
602
603
604      001000      START#1000      !START OF THE DISPLAY BUFFER.
605      007000      OLIMIT#7000      !APPROXIMATE END OF THE DISPLAY BUFFER.
606      007776      THPENO#7776      !LOCATION OF INITIALIZATION STACK.
607      000004      CORSTR#4      !LOCATION OF POP-11 TRAP VECTOR.
608      007012      JMPADDR#LIMIT+10; !WHERE THE POINTER IS TO FIRST CHAR ON SCREEN
609      000040      NUMLINE#32;      !NUMBER OF LINES ON TEXT TO SHOW ON THE SCREEN
610
611      005010      CRLF#5010      !CARRIAGE RETURN = LINE FEED
612      000175      ALTHOO#175      !THE "KEY" CHARACTER (I.E., ALTHOOES).
613
614      160000      OISJMP#160000      !THE GT48 JMP INSTRUCTION
615      173000      OISTOP#173000      !THE GT48 STOP DISPLAY INSTRUCTION.
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
      .SETTL  INITIALIZATION AND RESTART CODE
      .PAGE
  
```

```

636
637
638
639
640
641
642
643
644
645      006000      .#6000
646
647
648
649
650
651
652
653
654
655
656
657
658
659      001000  000000  000007  175610  START:  RESET      !RESET ALL HARDWARE NOW.
660      000002  012737  000000      MOV      #7,OL11IS      !INITIALIZE OL-11 INPUT NOW.
661      000010  012706  007776      MOV      #THPENO,SP      !ESTABLISH A GOOD TEMPORARY STACK
662
663      000014  005237  175614      INO      OL11OS      !POINTER FOR CORE SEARCH.
664      000020  004339  166002      JSR      SCANOUTLIT#160000      !SET BREAK BIT
665      000024  000000      .WORD 0      !FOR 2 CHARACTER TIMES
666
667      000026  012703  000004      MOV      #CORSTRISCAN      !GET ADDRESS OF BAD CORE TRAP VECTOR.
668      000032  012723  166042      MOV      #NOTHERE#160000,(SCAN); !AND INSERT A POINTER TO US THERE.
669
670      000036  005023      ENOCOR:  CLR      (SCAN);      !NOW CLEAR ALL OF MEMORY BEYOND THE POINTER.
671      000040  000776      BR      ENOCOR      !UNTIL WE RUN OUT OF MEMORY AND TRAP.
672
673
674      000042  005743      NOTHER:  TST      =(SCAN)      !WHEN WE TRAP OUT, WE COME HERE.
675
676
677
678
679      000044  010306      MOV      SCAN#SP      !WE BACK UP POINTER TO GOOD CORE.
680
681      000046  105737  175614      IS:      TSTB      OL11OS      !NOTE THAT IF WE TRAP OUT AGAIN, IT
682      000052  100378      IS      IS      !IS STILL OK, BECAUSE WE WILL LOOP
683      000054  005037  175614      CLR      OL11OS      !UNTIL WE GET A GOOD CORE ADDRESS.
684
685
686
687
688
689
      !WHEN WE GET ONE, THAT IS LAST LOCATION
      !IN THE MACHINE, AND HENCE OUR SP.
      !SEE IF BREAK IS ON
      !GO BACK
      !CLEAR BREAK BIT
  
```

```

698          |          RESTART INITIALIZATION CODE WHEN COMMUNICATIONS IS WORKING.
699          |          -----
700
701          |
702          |
703          |
704          |
705          |
706          |
707          |
708          |
709          |
710          |
711          |
712          |
713          |
714          |
715          |
716          |
717          |
718          |
719          |
720          |
721          |
722          |
723          |
724          |
725          |
726          |
727          |
728          |
729          |
730          |
731          |
732          |
  
```

698					RESTART INITIALIZATION CODE WHEN COMMUNICATIONS IS WORKING.
699					-----
700					
701					
702					
703					
704					
705					
706					
707					
708					
709					
710					
711					
712					
713					
714					
715					
716					
717					
718					
719					
720					
721					
722					
723					
724					
725					
726					
727					
728					
729					
730					
731					
732					

SETTL VTR SIMULATOR
 PAGE

```

733          |
734          |
735          |
736          |
737          |
738          |
739          |
740          |
741          |
742          |
743          |
744          |
745          |
746          |
747          |
748          |
749          |
750          |
751          |
752          |
753          |
754          |
755          |
756          |
757          |
758          |
759          |
760          |
761          |
762          |
763          |
764          |
765          |
766          |
767          |
768          |
769          |
770          |
771          |
772          |
773          |
774          |
775          |
776          |
777          |
778          |
779          |
780          |
781          |
782          |
783          |
784          |
785          |
786          |
  
```

733					
734					
735					
736					
737					
738					
739					
740					
741					
742					
743					
744					
745					
746					
747					
748					
749					
750					
751					
752					
753					
754					
755					
756					
757					
758					
759					
760					
761					
762					
763					
764					
765					
766					
767					
768					
769					
770					
771					
772					
773					
774					
775					
776					
777					
778					
779					
780					
781					
782					
783					
784					
785					
786					

VTR (SCROLLING) PORTION OF THE BOOTSTRAP

NXTCHR JSR PC GETCHR 160000
 CMP CNAR 0127
 BGE NXTENR
 CMP CHAR 040
 BGE NORMAL
 MOV CHAR SCAN
 SUB 07 SCAN
 CMP SCAN 07
 BHS NXTCHR
 ASL
 ADD SCAN PC

IF GET A CHARACTER NOW.
 IS IT OUT OF RANGE?
 IF YES, GET ANOTHER ONE.
 IS IT A PRINTING CHARACTER?
 IF YES, IT'S A NORMAL PRINTING CHARACTER.
 MOVE IT OVER SO WE CAN PLAY WITH IT.
 IF AS SO THAT BELL [7] IS ZERO.
 IF CHARACTER IS LESS THAN BELL OR
 GREATER THAN CR, THEN IGNORE.
 IF GOOD, MAKE IT WORD INDEX.
 AND GO TO THE CORRECT ROUTINE.

IF BELL
 IF BACKSPACE
 IF TAB
 IF LINE FEED [LF]
 IF VERTICAL TAB [VT]
 IF FORM FEED [FF]
 IF CARRIAGE RETURN [CR]

CR MOV 0-1 TABCNT
 IF FALL THROUGH TO INSERT THE CHARACTER.

NORMAL JSR PC INSERT 160000
 INC TABCNT
 BR NXTCHR

IF INSERT THE CHARACTER IN THE BUFFER.
 UPDATE TAB POSITION NOW.
 AND GET NEXT CHARACTER.

TAB MOV 040 CHAR
 JSR PC INSERT 160000
 INC TABCNT
 BTT 07 TABCNT
 BNE TAB
 BR NXTCHR

IF A TAB, INSERT BLANKS UNTIL THE
 NEXT CHARACTER POSITION IS A MULTIPLE
 OF 8.
 ARE WE DONE YET?
 IF NO,
 IF YES.

VT MOV 040 COUNTER
 IF THIS PUTS THE LOW BYTE OF THE
 BRANCH CODE IN COUNTER, SAVE A WORD

BELL CLR 040 BR
 BR NXTCHR

IF BELL - WRITE IN 040 BR
 AND LOOP BACK

```
787
788 004256 012705 000040      FFI      MOV      #NUMLIN:COUNTR      IFORN FEED IS DONE BY INSERTING LF'S.
789
790 004262 012708 000012      FFL00P1  MOV      #12:CHAR      MAKE THE CHARACTER A LINEFEED.
791 004266 004737 166304      JSR      PC,LFSUB:160000      DO A LINEFEED.
792 004272 005305      DEC      COUNTR      DONE?
793 004274 003372      BGT      FF,00P      NOPE, KEEP SENDING THEM.
794 004276 000715      BR      NXTCHR      YES, NOW RETURN, DO NOT FALL THROUGH.
795
796
797 004308 012746 166132      LFI      MOV      #NXTCHR:160000,--(SP)      IRETURN TO NXTCHR AFTER PROCESSING
798                                     ITHE LF BY MAKING A JSR,
799
800 004314 013703 007012      LFSUB1  MOV      JHPADD:SCAN      IGET POINTER TO FIRST CHAR ON SCREEN
801
802 004318 122308      LFL00P1  CWPB      (SCAN)*:CHAR      IAND LOOK FOR A LINEFEED.
803 004322 001406      BEO      LFOUND      GOT IT, SEARCH HAS ENDED.
804 004324 020327 007000      CWP      SCAN:#BLIMIT      ARE WE AT END OF BUFFER?
805 004326 103773      BLO      LFL00P      NOPE, KEEP ON LOOKING.
806 004328 012703 001000      MOV      #START:SCAN      IF AT TOP, RESET TO BOTTOM OF BUFFER
807 004326 000770      OR      LFL00P      IAND KEEP ON LOOKING.
808
809 004330 005203      LFOUND1  INC      SCAN      IWE'VE GOT THE LINE FEED, STOP SHOWING
810                                     BIC      #1,SCAN      IFIRST LINE BY CHANGING THE "DISJMP"
811                                     MOV      SCAN:JHPADD      INSTRUCTION TO FIRST CHAR BEYOND P.
812 004342 004737 166350      JSR      PC,INSERT:160000      INSERT THE LF IN THE BUFFER.
813 004346 005000      CLR      CHAN      IAND THEN INSERT ONE NULL CHARACTER BECAUSE
814                                     ITHE "DISJMP" ADDRESS MUST BE EVEN, AND
815                                     ITWIS GUARANTEES HE WILL NOT LOSE A
816                                     IA GOOD DATA CHARACTER, WE FALL THROUGH
817                                     ITO INSERT THE NULL IN THE BUFFER.
818
819
820 004350 110021      INSERT1  MOVWB     CHAN:(POINTR)*      ISTICK IN THE CHARACTER NOW.
821 004352 032701 000001      BIT      #1,POINTR      IS NEXT POSITION EVEN OR ODD?
822 004356 001021      BNE      INSERTX      IODD, NO PROBLEMS, SPACE IS ALLOCATED.
823 004360 020127 007000      CWP      POINTR:#BLIMIT      EVEN, ARE WE AT THE END OF THE BUFFER?
824 004364 103410      BLO      INSERTL      NO, JUST MAKE ROOM FOR ANOTHER WORD.
825 004366 010103      MOV      POINTR,POINTR      IAT THE END, MOVE THE STUFF TO THE
826 004370 012701 001000      MOV      #START:POINTR      BEGINNING OF THE BUFFER.
827 004374 004737 166406      JSR      PC,INSERTL:160000      ICALL THE ROUTINE TO SAVE SPACE.
828 004378 005023      CLR      (SCAN)*      IAND CLEAR UP THE INSTRUCTIONS AT THE
829 004382 005013      CLR      (SCAN)      IEND OF THE BUFFER.
830 004384 000207      RTS      PC      IAND THEN RETURN.
831
832 004386 022121      INSERTL  CWP      (POINTR)*:(POINTR)*      IBYPASS THE "DISJMP" BY ADDING 4 TO POINTR,
833 004390 012711 166474      MOV      #HEADER:160000,(POINTR)      INOW INSERT THE DISJMP INSTRUCTION TO OUR HEADER
834 004394 012741 160000      MOV      #DISJMP:(POINTR)      IAND IT'S ADDRESS (PUT THEM IN BACKWARDS).
835 004398 005041      CLR      -(POINTR)      MAKE AVAILABLE A NEW CHARACTER SPOT.
836
837 004422 000207      INSERTX  RTS      PC      IFINALLY RETURN TO THE CALLER.
838
839
840
```

```
841
842
843 004424 012737 001000 172000  GT0USE1  MOV      #BSTART:GT48PC      ION A BUS ERROR, WE MERELY RESTART THE GT48 AT
844
845                                     ITHE P1 FOR THIS ROUTINE
846                                     ITIS THE FIRST WORD OF THE TABLE
847                                     IBELOW IT SAVES A WORD.
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864 004432 000002      SETUP1  .WORD      2      INITIALIZE 2 WORDS,--ALSO RTI FROM ABOVE
865 004434 000330      .WORD      330      ISTARTING AT LOCATION 330
866 004436 166424      .WORD      CTRUSE:160000      IFIRST WORD IS POINTER TO BUS ERROR ROUT
867 004440 000200      .WORD      200      ISECOND WORD IS NEW STATUS WORD ON INTERRUPT.
868
869 004442 000007      .WORD      7      INITIALIZE THE END OF THE BUFFER TO
870 004444 006776      .WORD      BLIMIT=0      IA CLEAR SPACE TO INSERT THE CHARACTER.
871 004446 000000      .WORD      0      ITHIS IS THE "RUNNING" START, THIS IS
872 004448 160000 166474      .WORD      DISJMP:HEADER:160000      IFOLLOWED BY A DISJMP TO OUR HEADER BLOCK
873 004450 160000 001000      .WORD      DISJMP:START      IAND THEN A DISJMP TO THE START OF THE BUFFER
874 004452 160000 006700      .WORD      DISJMP:BLIMIT=NUMLIN=NUMLIN      IAND A DISJMP TO THE FIRST CHAR ON SCREE
875
876 004464 000001      .WORD      1      IFINALLY START THE GT48 DOING AT
877 004466 172000      .WORD      GT48PC      ITHE POSITION INSTRUCTION IN THE
878 004470 166474      .WORD      HEADER:160000      IHEADER BLOCK.
879
880 004472 000000      .WORD      0      IEND OF INIT CODE
881
882
883
884 004474 103334      HEADER1  .WORD      103334      IENABLE CHAR MODE,BLINKING
885 004476 000177      .WORD      177      IA BLINKING BOX=BUS OUTI
886 004480 116124      .WORD      116124      IGO TO POINT MODE
887 004482 171340      .WORD      171340      ILOAD STATUS REGISTER
888 004484 000000 001352      .WORD      0,1352      IPOINT TO UPPER LEFT
889 004486 103324      .WORD      103324      IBACK TO CHAR MODE
890 004488 160000 007010      .WORD      DISJMP:JHPADD=2      IAND TO THE CHANGING JMP INST.
891
892
893
894
895
896
897
898
899
900
901
902
903
```

```

894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947

```

COMMUNICATIONS HANDLING ROUTINES

THE DL=11 HANDLER
 --- -- -- -- --

```

      GETOL: TSTB   DL11IS      ;CHECK THE HOST INPUT STATUS.
      BPL     GETOL1          ;HOST DID NOT SEND ANYTHING. YET.
      MOV     DL11IS,CHAR      ;HOST SENT US A CHARACTER. PROCESS IT.
      NOV     #7,DL11IS       ;REENABLE THE HOST TELECOMMUNICATIONS.
      BIC     #280,CHAR        ;MAKE CHARACTER JUST SEVEN BITS.
      BEQ     GETOL           ;IF NULL, IGNORE IT.
      RTS     PC              ;ELSE RETURN NOW.

      GETOL1: TSTB   KBD1S      ;DID USER TYPE A CHARACTER?
      BPL     GETOL           ;NO, GO BACK AND CHECK HOST MACHINE.
      MOV     KBD1S,DL1108     ;MOVE THE CHARACTER TO THE HOST.
      BR      GETOL           ;AND CHECK AGAIN FOR INPUT.

      GETCHR: JSR     PC,GETOL1160000 ;GET A CHARACTER FROM THE HOST NOW.
      CMP     CHAR,#ALTN00     ;IS IT AN "ALTN00"?
      BNE     GETEXT          ;NO, EXIT NOW.

      JSR     PC,GETOL1160000 ;YES, GET ANOTHER ONE NOW.
      CMP     CHAR,#'L'        ;IS IT AN "L"?
      BNE     LOADER          ;YES, START LOADING NOW.
      CMP     CHAR,#'R'        ;IS IT AN "R"?
      BNE     GETEXT          ;NO, IGNORE THE ALTN00 AND JUST RETURN THE CHAR.

      MOV     #01STOP,JMPAD=2  ;YES, REDEF, STOP DISPLAY BY INSERTING A "01STOP
      PRESTR: JMP     RES1RT1160000 ;INSTRUCTION IN THE BUFFER, AND RESTART.

```

```

948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001

```

THE "GET A SIX BIT CHARACTER" ROUTINE
 --- -- -- -- --

```

      GETSIX: JSR     PC,GETCHR1160000 ;GET A CHARACTER NOW.
      CMP     CHAR,#00        ;IS IT A LEGAL PRINTING CHARACTER?
      BNE     L,BAD           ;NOPE, ABORT
      CMP     CHAR,#137       ;IS IT BIG ENOUGH, IS IT TOO BIG?
      BNE     L,BAD           ;YEP, ABORT.

      RTS     PC              ;RETURN TO THE CALLER.

      OUTLIT: MOV     (SCAN)*10L100B ;THIS OUTPUTS TWO CHARACTERS VIA A
      MOV     (SCAN)*10L100B ;JSR SCAN,OUTLIT
      RTS     PC              ;TWO CHARACTERS.

      OUTLIT: MOV     (SCAN)*10L100B ;DOUBLE BUFFERED
      MOV     (SCAN)*10L100B ;RETURN

      GET8: JSR     PC,GETSIX1160000 ;GET A SIXBIT CHARACTER.
      MOV     L,BYT,*[SP]        ;SAVE IT ON THE STACK.
      TST     [INDEX]           ;UPDATE INDEX TO NEXT ITEM (ALL ARE #2)
      JNE     GETS8=21160000(INDEX) ;AND DISPATCH ACCORDING TO THE INDEX.

      GETS8: BR      GETS1      ;INDEX=2: ASSEMBLE FIRST CHAR
      BR      GETS2            ;INDEX=4: ASSEMBLE SECOND CHAR
      BR      GETS3            ;INDEX=6: ASSEMBLE THIRD AND LAST CHAR
      BR      GETS4            ;INDEX=8: RESET INDEX TO 2 (3) AND RETRY.

```

THIS ROUTINE DIFFERS FROM THE PREVIOUS ROUTINES
 IN THAT IT WILL TAKE SIX BIT CHARACTERS AND ASSEMBLE
 THEM FOR THE LOADER TO USE. NOTE THAT FROM THIS POINT
 ON WE WILL SWITCH TO THE LOADER DEFINITIONS OF THE
 REGISTERS, THUS THE CHARACTER IS RETURNED IN
 REGISTER "L,BYT" RATHER THAN CHAR (THOUGH THEY ARE
 PHYSICALLY THE SAME).

1022							
1023	001776	012783	000002	GET84	MOV	#2,INDEX	THE FOURTH INDEX IS THE SAME AS THE FIRST INDEX, JUST RESET IT AND FALL THROUGH.
1024							
1025							
1026							
1027	001712	004737	166630	GET81	JSR	PC,GETSIX:160000	GET ANOTHER CHARACTER NOW.
1028	001716	010004			MOV	L,BYT,HOLD	AND PRESERVE IT FOR NEXT TIME THROUGH.
1029	001720	006300			ABL	L,BYT	NOW THROW AWAY LEFT MOST BITS OF
1030	001722	006300			ASL	L,BYT	THE 8 BIT CHARACTER, NOW MERGE IN
1031	001724	106300			ASL	L,BYT	THE LEFT TWO BITS OF THE
1032	001726	106300			ROL	(BP)	IN SIX BIT CHARACTER WITH THE SIX
1033	001730	106300			ASL	L,BYT	BITS FROM THE CHARACTER ON THE
1034	001732	106116			ROL	(SP)	STACK, 1ST CHARACTER IS NOW ASSEMBLED.
1035	001734	012600			MOV	(SP)+,L,BYT	SO WE'LL RETURN IT TO THE USER.
1036	001736	000207			RTB	PC	AND THEN WE SHALL RETURN TO HIM.
1037							
1038							
1039	001740	006300		DET82	ASL	L,BYT	THE SECOND CHARACTER IS CREATED FROM
1040	001742	006300			ASL	L,BYT	THE 4 RIGHT BITS OF THE PREVIOUS CHARACTER
1041	001744	106300			ASL	L,BYT	AND THE FOUR KIDDE BITS OF THE PRESENT
1042	001746	106104			ROL	HOLD	8 BIT CHARACTER.
1043	001750	106300			ASL	L,BYT	WE WILL CREATE THE NEW 8 BIT
1044	001752	106104			ROL	HOLD	IN THIS REGISTER, SINCE IT
1045	001754	106300			ASL	L,BYT	MORE CONVENIENT, WE WILL MOVE OVER THE
1046	001756	106104			ROL	HOLD	ANSWER AT THE END.
1047	001758	106300			ASL	L,BYT	ONE MORE TO GO
1048	001762	106104			ROL	HOLD	DONE.
1049	001764	010400			MOV	HOLD,L,BYT	BRING OVER THE VALUE.
1050	001766	012604			MOV	(SP)+,HOLD	AND REMEMBER THE LAST CHARACTER WE RECEIVED.
1051	001770	000207			RTS	PC	AND RETURN TO THE CALLER.
1052							
1053							
1054	001772	006100		GET83	ROL	L,BYT	FINAL CHARACTER IS EASY, JUST A
1055	001774	106100			ROL	L,BYT	8-BIT MERGE OF LEFT TWO BITS OF
1056	001776	006004			ROL	HOLD	PREVIOUS VALUE WITH RIGHT SIX BITS
1057	001780	106000			ROL	L,BYT	OF LAST (4TH) CHARACTER RECEIVED.
1058	001782	006004			ROL	HOLD	
1059	001784	106000			ROL	L,BYT	AND WE ARE DONE.
1060	001786	005726			TST	(SP)+	FINALLY THROW AWAY STACK,
1061	001810	000207			RTS	PC	AND RETURN TO THE CALLER.
1062							
1063							
1064							
1065							
1066							
1067							
1068							
1069							
1070							
1071							
1072							
1073							
1074							
1075							
1076							
1077							
1078							
1079							
1080							
1081							
1082							
1083							
1084							
1085							
1086							
1087							
1088							
1089							
1090							
1091							
1092							
1093							
1094							
1095							
1096							
1097							
1098							
1099							
1100							
1101							
1102							
1103							
1104							
1105							
1106							
1107							
1108							
1109							

.SBYTL THE LOADER
.PAGE

1096							
1097							
1098							
1099							
1100							
1101							
1102							
1103							
1104							
1105							
1106							
1107							
1108							
1109							
1110							
1111							
1112							
1113							
1114							
1115							
1116							
1117							
1118							
1119							
1120							
1121							
1122	001154	004737	166664	L.PTR	JSR	PC,GETSIX:160000	ASSEMBLE AN 8 BIT CHARACTER NOW.
1123	001158	006005			ADD	L,BYT,L,CKSH	UPDATE THE CHECKSUM NOW.
1124	001162	005302			OEC	L,BYT	DECREMENT THE CHARACTER COUNTER.
1125	001164	000207			RTS	PC	AND RETURN TO THE CALLER NOW.
1126							
1127							
1128							
1129	001126	004737	167114	L.GHRO	JSR	PC,L,PTR:160000	ASSEMBLE A WORD, FIRST GET A CHARACTER

1110 007132 010046
1111 007134 004737 167114
1112 007140 000300
1113 007142 052600
1114 007144 000207
1115
1116
1117
1118
1119 007146 004737 167126
1120 007132 010046
1121 007134 004737 167114
1122 007100 109705
1123 007102 001346
1124
1125 007164 004337 166652
1126 007170 175 107
1127
1128 007172 032716 000001
1129 007176 001401
1130
1131 007200 000000
1132
1133 007202 000136
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150

MOV L,BYT,={SP}
JSR PC,L,PTB,160000
SWAB L,BYT
BIS {SP},L,BYT
RTS PC

!AND SAVE IT.
!AND THEN GET ANOTHER ONE.
!AND THEN REASSEMBLE THE MESS,
!WITH THE FEARSOME POWER OF THE 11.
!AND RETURN TO THE CALLER.

L.JMPI JSR PC,L,GNRO,160000
MOV L,BYT,={SP}
JSR PC,L,PTB,160000
TSTB L,CNRM
BNE L,BAD
JSR SCAN,OUT,IT,160000
L,BYTE ALTHOO,IC
B,YT #1,SP
BEO L,JMP1
L.WALT, WALT
L.JMP1, JMP #0,PT1

!ALL DONE WITH THE LOAD. ASSEMBLE
!THE STARTING ADDRESS NOW.
!AND DON'T FORGET TO CHECKSUM IT.
!A BAD CHECKSUM. ALL IS EVIL.

!GOOD CHECKSUM. INFORM HOST
!WITH ALTHOO G

!DO WE WANT TO START EXECUTION?
!YES, AWAY WE GO.

!IF NOT, WALT.

!IF GO, THEN GO ALREADY. WHEEEE!

.SBTTL THE SELF TEST

.PAGE

1151
1152
1153
1154
1155 100000
1156 104000
1157 110000
1158 114000
1159 120000
1160 124000
1161 130000
1162
1163 002000
1164 002200
1165 002400
1166 002600
1167 003000
1168 003200
1169 003400
1170 003600
1171
1172 000100
1173 000140
1174 000020
1175 000030
1176
1177 000004
1178 000005
1179 000006
1180 000007
1181
1182 160000
1183 164000
1184 170000
1185 173400
1186
1187 000300
1188 000200
1189 000040
1190 000000
1191 000004
1192
1193
1194 174000
1195
1196 000100
1197 040000
1198 001777
1199 001777
1200 020000
1201 020000
1202 017000
1203 000077
1204 000100

CHAR=100000
SHORTV=104000
LONGV=110000
POINT=114000
GRAPHX=120000
GRAPHY=124000
RELATV=130000
INT0=2000
INT1=2200
INT2=2400
INT3=2600
INT4=3000
INT5=3200
INT6=3400
INT7=3600

LPOFF=100
LPON=140
BLKOFF=20
BLKON=30

LINE0=4
LINE1=5
LINE2=6
LINE3=7

OJMP=100000
ONOP=104000
STATSA=170000
OSTOP=173400

LPLITE=300
LPDARK=200
ITAL0=40
ITAL1=00
SYNQN=0

STATSB=174000

INCR=100
INTX=40000
MAXX=1777
MAXY=1777
MINUSX=20000
MINUSTHINUSX
MAXSX=17000
MAXSY=77
MINUSY=100

!THIS IS GT48 QUICK TEST
!GIVES QUICK VISUAL TEST
!OF CONDITION OF MACHINE
!WITHOUT READING IN OIAG.

!BRIGHTEST

!STOP INTERRUPT

!ITALICS OFF
! " ON
!SYNCR ON

!LOAD GRAPH INCR
!INTERRUPT BIT
!BIGGEST X VECTOR
!BIGGEST Y VECTOR
!THE MINUS BIT

!BIGGEST X IN SHORTVEC
! " Y IN "
!MINUS BIT FOR ! IN SHORTVEC

1205						
1206						
1207	007284	012737	167214	172000	MOV #FILE01160000,GT40PC	START THE GT40
1208	007212	000001			WAIT	AND WAIT
1209						
1210	007214	114000			FILE01 POINT1BLKOFF	POINT--INVISIBLE
1211	007216	000000			0	
1212	007220	001377			MAXY	
1213						
1214	007222	112004			LONGVINT01LINE0	ORAW TOP LINE
1215	007224	041777			INTXIMAXX	
1216	007226	000000			0	
1217						
1218	007230	112400			LONGVINT21LINE1	
1219	007232	040000			INTX	ORAW LINE TO RIGHT
1220	007234	021377			MINUSXIMAXY	
1221						
1222	007236	113006			LONGVINT41LINE2	
1223	007240	061777			INTXIMINUSXIMAXX	ORAW BOTTOM LINE
1224	007242	000000			0	
1225						
1226	007244	113407			LONGVINT61LINE3	
1227	007246	040000			INTX	
1228	007250	001377			MAXY	ORAW LINE TO LEFT
1229						
1230	007252	114000			POINT	
1231	007254	000400			400	
1232	007256	000500			500	
1233	007260	106200			SHORTVINT1	
1234	007262	057677			57677	I=X+Y
1235	007264	106600			SHORTVINT3	
1236	007266	077677			77677	I=X+Y
1237	007270	107200			SHORTVINT5	
1238	007272	077777			77777	I=X+Y
1239	007274	107600			SHORTVINT7	
1240	007276	057777			57777	I=X+Y
1241						
1242	007300	114000			POINT	
1243	007302	001400			1400	
1244	007304	000500			500	
1245	007306	133030			RELATIVINT41BLKON	
1246	007310	057677			57677	I=X+Y
1247	007312	077677			77677	I=X+Y
1248	007314	077777			77777	I=X+Y
1249	007316	057777			57777	I=X+Y
1250						
1251	007320	114000			POINT	
1252	007322	000400			400	
1253	007324	000100			100	
1254	007326	174120			STATSBINCR20	TRY GRAPH MODES
1255	007330	114000			POINT	
1256	007332	001000			1000	
1257	007334	000200			200	
1258						

1259	007336	120000			GRAPHX	
1260	007340	001010			1010	
1261	007342	001020			1020	
1262	007344	001030			1030	
1263	007346	001040			1040	
1264	007350	001050			1050	
1265						
1266	007352	114000			POINT	
1267	007354	001000			1000	
1268	007356	001200			1200	
1269						
1270	007360	124000			GRAPHY	
1271	007362	001020			1020	
1272	007364	001030			1030	
1273	007366	001040			1040	
1274	007370	001050			1050	
1275	007372	001060			1060	
1276						
1277	007374	160000			QJMP	
1278	007376	167214			FILE01160000	
1279						
1280					.SBTTL PAPER TAPE BOOT	

```

1282
1283
1284
1285      177550
1286      177550
1287
1288
1289 001400 012701 160000
1290 001404 012702 000004
1291 001408 012703 167500
1292 001412 012712
1293 001416 012706 000024
1294 001420 014304
1295 001424 005714
1296 001428 100775
1297 001432 010712
1298 001436 012706 000024
1299 001440 010441
1300
1301 001444 040601
1302 001448 010111
1303 001452 011102
1304 001456 005214
1305 001460 105714
1306 001464 100376
1307 001468 114412 000002
1308 001472 005211
1309 001476 120227 000375
1310 001480 001366
1311 001484 105222
1312 001488 000142
1313
1314
1315
1316 001474 177550
1317 001476 177550
1318
1319

```

; PAPER TAPE BOOT
 ; HIGH SPEED READER ADDRESS
 ; LOW SPEED READER ADDRESS
 ; ORIGIN=1400
 ; SET MEMORY CHECK LIMITS
 ; TRAP ADDRESS IS LOC. 4
 ; POINTER TO DEVICE ADDRESSES
 ; PRESET TRAP ADDRESS IN LOC. 4
 ; STACK SET UP AT SPECIAL ADDRESS
 ; GET DEVICE ADDRESS
 ; CHECK AVAILABILITY OF DEVICE
 ; CHECK DEVICE FOR ERRORS
 ; RESET TRAP ADDRESS AT LOC. 4
 ; SPECIAL ADDRESS USED AS MASK LATER
 ; DO MEM CHK READER STATUS ADDRESS
 ; IS MOVED
 ; SET R1=X7752, MASK IN SP=24
 ; STORE OWN ADDRESS IN POINTER
 ; GET BYTE POINTER
 ; ENABLE READER
 ; TEST DONE BIT
 ; WAIT UNTIL READY
 ; THEN PICK IT UP AND STORE IT
 ; BUMP POINTER
 ; STORED JUMP OFFSET
 ; NOT YET
 ; YES, ALL DONE
 ; GO EXECUTE AS BRANCH
 ; DEVICE ADDRESSES FOLLOW - DO NOT CHANGE THE ORDER
 ; LOW SPEED READER
 ; HIGH SPEED READER
 ; SBTTL CASSETTE BOOT

```

1322
1323
1324
1325      177550
1326
1327 001500 012700 177500
1328 001504 005010
1329 001508 010701
1330 001512 062701 000052
1331 001516 012702 000375
1332 001520 112103
1333
1334 001524 112110
1335 001528 100413
1336 001532 130310
1337 001536 001776
1338 001540 105202
1339 001544 100772
1340 001548 116012 000002
1341 001552 120337 000000
1342 001556 001767
1343 001560 000000
1344 001564 000755
1345
1346 001568 005710
1347 001572 100774
1348 001576 005007
1349
1350 001580 017640
1351
1352 001584 002415
1353
1354 001588 112024
1355
1356 001592 000000 000000
1357 001596 167500
1358 001600 000340
1359
1360
1361

```

; CASSETTE BOOT
 ; ITA-11 CONTROL AND STATUS REGISTER
 ; ORIGIN=1000
 ; SELECT UNIT #0
 ; USE FOR PIC
 ; R1 HOLDS ADDR. OF COMMAND TABLE
 ; MEMORY PTR. AND DATA FLAG
 ; TEST BITS
 ; COMMAND FROM TABLE TO TACS
 ; WHEN COMMAND DONE NEC. QUIT
 ; TEST READY AND T-REQ BITS IN TACS
 ; LOOP 'TIL SOMETHING COMES UP
 ; ADVANCE MEMORY POINTER
 ; IF MINUS, TRY NEXT COMMAND
 ; READ DATA INTO MEMORY
 ; FIRST BYTE READ SHOULD BE '240'
 ; IF O.K., GO READ ANOTHER BYTE
 ; HALT ON ERROR
 ; RESTART ON CONTINUE
 ; CHECK FOR ERROR
 ; HALT ON ERROR
 ; JMP #02
 ; .BYTE 240: READY=T-REQ,
 ; .BYTE 37: ILDS=READY=0
 ; .BYTE 15: SFB=GO
 ; .BYTE 5: READ=GO
 ; .BYTE 24: READ=ILDS
 ; .BYTE 224: READ=ILDS=0, TABLE
 ; THESE ARE FILLER WORDS
 ; POINTER UP VECTOR AND PRIORITY
 ; SBTTL MR13=00 BOOT

1363
1364
1365
1366
1367 007600 010702
1368 007602 000451
1369 007604 177402
1370 007606 000005
1371
1372 007610 010702
1373 007612 000445
1374 007614 177406
1375 007616 000005
1376
1377
1378 007620 010702
1379 007622 000417
1380 007624 177344
1381 007626 000005
1382 007630 000003
1383 007632 100000
1384 007634 024000
1385
1386
1387 007636 010702
1388 007640 000410
1389 007642 172524
1390 007644 000003
1391 007646 000011
1392 007650 000020
1393 007652 100000
1394
1395
1396 007654 010702
1397 007656 000423
1398 007660 176716
1399
1400
1401 007662 000005
1402 007664 010200
1403 007666 005720
1404 007670 012001
1405 007672 005311
1406 007674 005720
1407 007676 012041
1408 007680 031011
1409 007682 001776
1410 007684 005720
1411 007686 031041
1412 007690 001406
1413 007692 000112
1414
1415
1416 007714 167600

MR11-DB BULK STORAGE PROGRAM LOADER LISTING
;ORIGIN=1600 ;KEEP TRACK OF DRIDIN
RF11: MOV PC,R2 ;FIXED HEAD DISK (256 KW)
BR OTHER
177402
5
RK11: MOV PC,R2 ;MOVING HEAD DISK (DARTMOUTH)
BR OTHER
177406
5
TC11: MOV PC,R2
BR TAPES
177344
5 ;ADDRESS OF WORD COUNT
4003 ;LAST COMMAND
100000 ;FIRST COMMAND
24000 ;DONE MARK
;ERROR MARK
TM11: MOV PC,R2
BR TAPES
172524
00003 ;ADDRESS OF BYTE COUNT
00011 ;LAST COMMAND
200 ;FIRST COMMAND
100000 ;DONE MARK
;ERROR MARK
RP11: MOV PC,R2 ;MOVING HEAD DISK (PACK)
BR OTHER
176716
TAPES: RESET
MOV R2,R0 ;GET THE ADDRESS OF THE BRANCH
TST (R)+ ;R0 TO POINT AT LAST COMMAND
MOV (R)+,R1 ;GET THE WORD COUNT ADDRESS
DEC (R) ;SET UP FOR ADVANCE 1 RECORD
TST (R)+ ;MOVE R0 TO FIRST COMMAND
MOV (R)+,(R) ;COMMAND WORD TO COMMAND REG.
BIT (R),(R) ;LOAD FOR DONE INDICATORS
BCD ,=2 ;DONE SET, TRY AGAIN
TST (R)+ ;DONE FIRST COMMAND, CHECK FOR ERROR
BIT (R),(R) ;LOAD FOR SET ERROR BITS
BCD OTHER ;AND ERRORS = TRY THE READ
AGAIN: JMP (R) ;RERUN FOR ERRORS
RPVEC: RF11(160000) ;RF11 POWER UP VECTOR

1417 007716 000340
1418
1419 007720 010702
1420 007722 000401
1421 007724 177400
1422
1423
1424 007726 000005
1425 007730 010200
1426 007732 005720
1427 007734 010001
1428 007736 012711 177000
1429 007742 011041
1430 007744 032711 100200
1431 007750 001775
1432 007752 100757
1433 007754 005007
1434
1435 007756 000000
1436 007760 167610
1437 007762 000340
1438 007764 167720
1439 007766 000340
1440 007770 167654
1441 007772 000340
1442 007774 167620
1443 007776 000340
1444
1445
1446

340
RC11: MOV PC,R2 ;FIXED HEAD DISK (64KW)
BR OTHER
177400
;ADRS OF WORD COUNT (COMMAND+2)
;COMMAND WORD (3) IS THE RESET
OTHER: RESET
MOV R2,R0 ;R0 TO POINT AT WORD COUNT ADRS
TST (R)+ ;POINT TO ADDRESS
MOV (R)+,R1 ;WORD COUNT ADDRESS TO R1
MOV #-1000,(R) ;LOAD WORD COUNT
MOV (R),(R) ;COMMAND TO COMMAND REGISTER
BIT #100200,(R) ;CHECK FOR ERROR OR DONE
BCD ,=4 ;IF NEITHER, KEEP LOOKING
BHI AGAIN ;ERROR, TRY AGAIN
CLR PC
0 ;FILLER
RKVEC: RK11(160000) ;RK POWER UP VECTOR
RCVEC: RC11(160000) ;RC POWER UP VECTOR
RPVEC: RP11(160000) ;RP POWER UP VECTOR
TCVEC: TC11(160000) ;TC11 POWER UP VECTOR
;SBTTL ROM VERSION 1 VALUES
;PAGE

```

09AB1  AHA
DATA PATTERN STORED IN THE VT48 BOOTSTRAP VERSION 1
***** THIS IS A IMAGE LISTING OF THE VT48 5140 BOOTSTRAP *****
THE DATA IS A MIRROR IMAGE OF THAT IN THE BOOTSTRAP AS
ONLY THE ADDRESS FIELD IS CHANGED
BOOTVT.889  5/2/72    <SPECIAL>

VT-48 BOOTSTRAP LOADER, VERSION 889, RELEASE R81, 5/2/72
COPYRIGHT 1972; DIGITAL EQUIPMENT CORPORATION,
140 MAIN STREET
MAYNARD, MASSACHUSETTS
      B1754

WRITTEN BY JACK BURNES, SENIOR SYSTEMS ARCHITECT

THIS ROUTINE IS INTENDED TO BE LOADED IN THE ROM PORTION OF THE VT-48.

REGISTER DEFINITIONS:

R0=X0
R1=X1
R2=X2
R3=X3
R4=X4
R5=X5
R6=X6
R7=X7

SP=R6
PO=R7

RET1=R0
INP1=R1
INP2=R2
WORK1=R3
WORK2=R4
SCR1=R5

LOKSM=WORK1
LBYT=RET1
LBC=SCR1
LAOR=INP1

IRETURN OF VALUE REGISTER,
IARGUMENT FOR CALLED FUNCTION
ISECOND ARGUMENT,
IFIRST WORK REGISTER,
ISECOND WORK REGISTER,
ISCRATCH REGISTER,

IOVERLAPPING DEFINITIONS FOR LOADER PORTION.

```

```

DOREND=36000
ROMORG=160000

STARTX=0
STARTY=1360

VT40=C=172000-130000
KB01S=27360
P100B=25614
P101B=25610

KB01B=KB01S+2
P101B=P101S+2
P100B=P100S+2

P1000=COREND-2*10000
P101C=P1000+4
STKERT=P101C-2*38000

JMPDIS=160000

PWRPAL=24

;=16000
;=ROMORG

;SET THE ORIGIN NOW!!!

START: MOV     $PWRPAL+2,SCR1
      CLR     @SCR1
      MOV     PC,=@SCR1

      RESET

      ;PICK UP POINTER TO P.F. STATUS.
      ;CLEAR IT OUT TO BE SURE.
      ;SET UP THE RESET LOCATION.

      ;RESET THE BUS.

```

1555	016022	012767	000007	007570	MOV	#7,P0IS	INITIALIZE POP-10 INPUT	
1556	016000	012767	000001	011532	MOV	#1,R001	INITIALIZE TTY INPUT	
1558	016026	012767	000201	007500	MOV	#20,P100S	INITIALIZE POP-10 OUTPUT	
1559								
1560								
1561								
1562	016034	012706	015770		RSTRTI	MOV	#STKRTI,SP	ISSET UP THE STACK NOW!
1563	016040	005001			CLR	LAOR		ICLEAR ADDRESS POINTER
1564	016042	012702	160000		MOV	#JWP015,INP2	PLACE A DISPLAY JUMP INSTRUCTION IN A REGISTER.	
1565	016046	010221			MOV	INR0,(LAOR0)	MOVE IT TO LOCATION 0	
1566	016050	012711	166756		MOV	#01,PRG0150000,(LAOR)	MOVE ADDRESS POINTER INTO 2.	
1567	016054	012701	000030		MOV	#PWFAL*4,LAOR	ISSET UP WHERE WE WILL STORE CHARACTERS.	
1568	016060	005000			CLR	RET0	PREPARE TO INSERT A ZERO CHARACTER.	
1569	016002	004767	000022		JSR	PC,00CHAR	INSERT IT NOW.	
1570	016006	005067	003706		CLR	VT40PC	ICLEAR THE DISPLAY PROGRAM COUNTER AND START.	
1571								
1572	016072	004767	000210		MAJORI	JSR	PC,0TCHR	JO0 A CHARACTER NOW.
1573	016076	000240			NOP			
1574	016100	000240			NOP			
1575	016102	000240			NOP			
1576	016104	012746	166072		MOV	#MAJOR*150000,(BP)	INSERT IN DISPLAY BUFFER NOW.	
1577								
1578	016110	010105			00CHAR	MOV	LAOR0,BCR1	JO0 CURRENT BUFFER POSITION NOW.
1579	016112	022523			CMF	{SCR1}*{SCR1}	IBYPASS CURRENT DISPLAY JUMP.	
1580	016114	005025			CLR	{SCR1}	ICLEAR FUTURE ADDRESS FOR JUMP.	
1581	016116	010225			MOV	INR0,{BCR1}	ISTICK IN TEMPORARY JUMP WHILE WE REPLACE CURRENT	
1582	016120	005015			CLR	{SCR1}	IA DISPLAY JUMP TO ZERO.	
1583	016122	005011			CLR	{LAOR}	INOW REPLACE CURRENT DISPLAY JUMP BY THE CHARACT	
1584	016124	0050021			RTS	RET0,(LAOR)	IT'S DONE THIS WAY TO WASTE 2 CYCLES.	
1585	016126	010211			MOV	INR0,(LAOR)	ITO AVOID TIMING PROBLEMS WITH THE VT40.	
1586	016130	000207			RTS	PC	IANO FINALLY RETURN.	
1587								
1588								
1589								
1590								
1591								
1592								
1593								
1594								
1595								
1596								
1597								
1598								
1599								
1600								
1601								
1602								
1603	016132	004707	000124		GT8I	JSR	PC,RTSIX	JO0 SIX BITS NOW.
1604	016136	010046			MOV	RET0,(BP)	ISAVE THE CHARACTER NOW.	
1605	016140	000401			OR	GTP04	IBYPASS THE 8100	
1606	016142	005002			GT84I	CLR	INP0	IRESET THE MAGIC REGISTER NOW.
1607	016144	005722			GTP84I	TST	{INP2}	INCREMENT WHERE TO GO.
1608	016146	066207	166250		A00	OTS08*150000{INP2},PC	UPDATE PC NOW.	

Address	Hex	Assembly	Comments
1609	016152		
1611			
1612	016152 000104	GTSPH, 0T811 JSR PC,RTSIX	IS A CHARACTER NOW.
1613	016156 010004	MOV RET1,WORK2	ISAVE FOR A SECOND,
1614	016160 006300	ASL RET1	
1615	016162 006300	ASL RET1	ISWIFT TO LEFT OF BYTE
1616	016164 106300	ASLB RET1	
1617	016166 106116	ROLB #8P	IPACK THEM IN.
1618	016170 106300	ASLB RET1	
1619	016172 106116	ROLB #8P	IA GOOD 8 BIT THING.
1620	016174 012600	MOV (SP)+,RET1	IPCP AND RETURN NOW.
1621	016176 000207	RTS PC	
1622			
1623	016200 006300	GT821 ASL RET1	INORST CASE, SHIFT 1
1624	016202 006300	ASL RET1	
1625	016204 106300	ASLB RET1	
1626	016206 106104	ROLB WORK2	
1627	016210 106300	ASLB RET1	
1628	016212 106104	ROLB WORK2	
1629	016214 106300	ASLB RET1	
1630	016216 106104	ROLB WORK2	
1631	016220 106300	ASLB RET1	
1632	016222 106104	ROLB WORK2	
1633	016224 010400	MOV WORK2,RET1	
1634	016226 012604	MOV (SP)+,WORK2	
1635	016228 000207	RTS PC	
1636			
1637	016252 006100	GT831 ROL RET1	
1638	016254 006100	ROL RET1	
1639	016256 006004	ROL WORK2	
1640	016260 106000	ROLB RET1	
1641	016262 006004	ROLB WORK2	
1642	016264 106000	ROLB RET1	IFINAL CHARACTER ASSEMBLED.
1643	016266 005226	TST (SP)+	IFUOGE STACK,
1644	016268 000207	RTS PC	IANO RETURN NOW.
1645			
1646	016250	GT8T0 " ==2	IPUSH ZERO CONDITION BACK INTO NEVER-NEVER LANE.
1647			
1648	016252 000000	,HORO GT81=GT8P	
1649	016254 000026	,HORO GT80=GT8P	
1650	016256 000000	,HORO GT83=GT8P	
1651	016200 177770	,HORO GT82=GT8P	
1652			
1653			
1654	016262 004767 000020	OTSIX, JSR PC,RTCHR	
1655	016266 020027 000040	CHP RET1,#40	
1656	016272 002546	LBAD	
1657	016274 020027 000137	CHP RET1,#137	
1658	016300 003143	LBAD	
1659	016302 000207	RTS PC	
1660			
1661			
1662			

1663	01E384	005726		GTCHP1	TST	(SP)*	UPDATE THE STACK,
1664	01E386	012900	013772	GTCHM1	MOV	SP10=0000;RET1	SET UP POINTER TO THE INPUT CHARACTER.
1665	01E388	004767	000064	GTCHL1	JSR	PC,CHECK	
1666	01E38A	005710			TST	RET1	ANY CHARACTERS THERE?
1667	01E38C	001774			BEQ	GTCHL1	
1668	01E38E	011846			MOV	RET1,=SP	PUSH THE CHAR ON THE STACK,
1669	01E390	003020			CLR	(RET1)*	ICLEAR THE CHAR GOT TO NOW.
1670	01E392	042716	177600		BIC	#200,(SP)	ICLEAR AWAY PARITY NO.
1671	01E394	001784			BEQ	GTCHP1	IF ZERO, GET ANOTHER
1672	01E396	022716	000177		CHP	#177(00)	
1673	01E398	001761			BEQ	GTCHP1	ALSO IGNORE RUNOUTS.
1674	01E39A	022710	000175		CHP	#175,RET1	WAS IT A "175"
1675	01E39C	011618			MOV	(SP),RET1	NOPE.
1676	01E39E	000122			CHP	RET1,#122	IF, RESET IN CASE OF ABORT.
1677	01E3A0	000114			BEQ	RSTR	IS IT AN R
1678	01E3A2	000114			CHP	RET1,#114	IF, RESTART
1679	01E3A4	000145			BEQ	LOAD	IS IT AN L
1680	01E3A6	011610					IF, LOAD.
1681	01E3A8	012600		GTNP1	MOV	(SP),RET1	HOW DO THE FOCUSING,
1682	01E3AA	000175			MOV	(SP),RET1	
1683	01E3AC	000175			CHP	RET1,#175	
1684	01E3AE	000175			BEQ	GTCHP1	IF ALTHOGE, LOOP
1685	01E3B0	000175			RTS	PC	
1686	01E3B2	000175					
1687	01E3B4	000175					
1688	01E3B6	000175					
1689	01E3B8	000175					
1690	01E3BA	000175					
1691	01E3BC	000175					
1692	01E3BE	000175					
1693	01E3C0	000175					
1694	01E3C2	000175					
1695	01E3C4	000175					
1696	01E3C6	000175					
1697	01E3C8	000175					
1698	01E3CA	000175					
1699	01E3CC	000175					
1700	01E3CE	000175					
1701	01E3D0	000175					
1702	01E3D2	000175					
1703	01E3D4	000175					
1704	01E3D6	000175					
1705	01E3D8	000175					
1706	01E3DA	000175					
1707	01E3DC	000175					
1708	01E3DE	000175					
1709	01E3E0	000175					
1710	01E3E2	000175					
1711	01E3E4	000175					
1712	01E3E6	000175					
1713	01E3E8	000175					
1714	01E3EA	000175					
1715	01E3EC	000175					
1716	01E3EE	000175					

1717	01E502	002767	177400	027262	BIS	#400,P101C	MAKE SURE IT'S NONE ZERO;
1718	01E504	012767	000007	007072	MOV	#7,P101B	REINITIALIZE COMMUNICATION LINE.
1719	01E506	000207			CHECK48	RTS	AND RETURN.
1720	01E508	000207					
1721	01E50A	000207					
1722	01E50C	000207					
1723	01E50E	000207					
1724	01E510	000207					
1725	01E512	000207					
1726	01E514	000207					
1727	01E516	000207					
1728	01E518	000207					
1729	01E51A	000207					
1730	01E51C	000207					
1731	01E51E	000207					
1732	01E520	000207					
1733	01E522	000207					
1734	01E524	000207					
1735	01E526	000207					
1736	01E528	000207					
1737	01E52A	000207					
1738	01E52C	000207					
1739	01E52E	000207					
1740	01E530	000207					
1741	01E532	000207					
1742	01E534	000207					
1743	01E536	000207					
1744	01E538	000207					
1745	01E53A	000207					
1746	01E53C	000207					
1747	01E53E	000207					
1748	01E540	000207					
1749	01E542	000207					
1750	01E544	000207					
1751	01E546	000207					
1752	01E548	000207					
1753	01E54A	000207					
1754	01E54C	000207					
1755	01E54E	000207					
1756	01E550	000207					
1757	01E552	000207					
1758	01E554	000207					
1759	01E556	000207					
1760	01E558	000207					
1761	01E55A	000207					
1762	01E55C	000207					
1763	01E55E	000207					
1764	01E560	000207					
1765	01E562	000207					
1766	01E564	000207					
1767	01E566	000207					
1768	01E568	000207					
1769	01E56A	000207					
1770	01E56C	000207					
1771	01E56E	000207					
1772	01E570	000207					
1773	01E572	000207					
1774	01E574	000207					
1775	01E576	000207					
1776	01E578	000207					
1777	01E57A	000207					
1778	01E57C	000207					

```

LPT1: JSR PC,GTB          IGT 8 BITS NOW.
      ADD LBY1,FLCKSM     IUPDATE OMECKSUM
      BIC #177000,LBY1   ICLEAN UP THE BYTE NOW.
      DEC LBC             IUPDATE THE COUNTER.
      RTS PC              IRETURN NOW.

LGWR0: JSR PC,LPTR        IGT A CHARACTER,
      MOV LBY1,=(SP)      ISAVE FOR A SECOND.
      JSR PC,LPTR         IGT ANOTHER CHARACTER,
      SWAB LBY1           INOW ASSEMBLE THE WORD.
      BIS (SP)+,LBY1      IAND RETURN WITH A 16 BITER.
      RTS PC

LJMP1: JSR PC,LGWR0       IGT A WORD
      MOV LBY1,=(SP)      ISAVE ON THE STACK,
      JSR PC,LPTR         IGT A CHARACTER.
      TSTB LCKSM          IS IS ZERO?
      BNE LBAO            IYEP. WHAT CRAP.
      BIT #1,ISPI         IS IT 000?
      BEQ LJMP1           IYEP. START PROGRAM GOING NOW.
      MOV (PC)+,RET1      ITELL FOR=10 WE'VE LOADED OK.
      .BYTE 175,107       I"CTRL GOOD"
      .BYTE 107,175       I"0000 CTRL"
      JSR PC,SENDIT
      HALT
      BR 1=2

LJMP1: JMP 0(SP)+        IAND AWAY WE GO.

SENDIT: JSR PC,CHECK      IPOLL THE OUTPUT DEVICE NOW.
      TST P100C           IOUTPUT CLEAR?
      BNE 3ENGIT         INOPE. LOOP AWHILE LONGER.
      MOV RET1,p100B      ISEND OUT THE CHARACTER.
      CLRB RET1           ICLEAR THE BYTE.
      SWAB RET1           IAND SWAP THEM NOW.
      BNE SENDIT         IIF NOT EQUAL, REPEAT.
      RTS PC

```

```

THIS IS THE INITIALIZING V440 PROGRAM WHICH WILL
JUMP TO THE PROGRAM AFTER THE POWER FAIL LOCATIONS
WHICH WILL JUMP TO ZERO WHICH WILL JUMP BACK TO HERE,

DISPRG# .WORD 170000
        .WORD 110000
        .WORD STARTX
        .WORD STARTY
        .WORD 100000
        .WORD JHPPDIS
        .WORD PHREAL**
        .WORD 0
        .WORD 0

        ILOAD STATUS REGISTER FOR NORMAL OPERATION
        ISET POINT MODE, "NORMAL",
        IX 00000000
        IY 00000000
        ISET CHARACTER MODE,
        ITHEN JUMP TO THE 'POWERFAIL' LOCATION,
        ITO DISPLAY USERS CHARACTERS,

        .END

```

AGAIN = 007712	ALYMOD = 000175	BELL = 000250	BLIMIT = 007000
RLNOFF = 000020	BLKON = 000030	BSTART = 001000	CHAR = 100000
CHARA = 001020	CHECK = 016402	CHECK1 = 010430	CHECK2 = 016450
CHEGK3 = 016466	CHECK4 = 016516	CORENO = 000000	CORSTR = 000004
COUNTR = 000000	CR = 000200	CRLF = 000010	DEV = 007476
DEV1 = 007422	DISJMP = 160000	DISPLA = 177570	DISPRG = 007556
DISTCP = 173000	DUMP = 160000	OL11IB = 170612	OL11IS = 170610
OL11CB = 175610	DL11DS = 175614	ONDP = 160000	OOCHAR = 016110
ONE = 007554	DOONF0 = 001430	ONE1 = 001502	OSR = 001000
OSTOP = 173400	DUMP = 001016	O2BTVP = 000144	ENO = 001416
FNCCCR = 006036	ERROR1 = 001130	ERROR2 = 001176	FF = 000254
FFLOOP = 006262	FILCNT = 001012	FILE0 = 007214	FILLER = 001010
GETCHR = 006564	GETOL = 006516	GETOL1 = 000546	DETEXT = 000650
GETSIX = 006630	GETB = 006664	GET8TB = 000700	DETS1 = 000712
GET82 = 006740	GETY3 = 006772	GET84 = 000706	DRAPMX = 120000
GRAPHY = 124000	GTBUDE = 006424	GTCHL = 000312	OTOHP = 006306
GTCHR = 006306	GNP = 006366	GT84 = 000144	OTBIX = 006260
GT40PC = 172000	GT40SR = 172002	GT8 = 000132	OT8P = 006192
GT8TB = 006250	GT01 = 006152	GT2 = 000200	OT83 = 006232
GT84 = 006142	HEADER = 000474	MOLO = 000004	WSR = 177550
ICNT = 001014	ICOUNT = 002124	IMAGE = 001004	INCR = 000102
INDEX = 000003	INP1 = 000001	INP2 = 000002	INSERT = 000350
INSRTL = 006406	INSRTX = 006422	INTX = 000000	INT0 = 000200
INT1 = 002200	INT2 = 002400	INT3 = 002600	INT4 = 003000
INT5 = 003200	INT6 = 003400	INT7 = 003600	ITAL0 = 000040
ITAL1 = 000460	JMPA00 = 000702	JMP01S = 160000	KBOIB = 002762
KBOIS = 002760	LAOR = 000001	LBA0 = 000010	LBC = 000000
IBYT = 000000	LCKSH = 000003	LF = 000300	LFLOOP = 006310
IFOUND = 006330	LFSUB = 006304	LGWR0 = 000646	LINE0 = 000004
LINE1 = 000004	LINE2 = 000006	LINE3 = 000007	LJMP = 006666
JMP1 = 006726	LLO2 = 006532	LLO3 = 000376	LLO4 = 006624
LOAD = 006920	LOADER = 007012	LONGV = 110000	LOOP = 007444
LOP1 = 007722	LDDP2 = 007526	LDPARK = 000200	LPLITE = 000300
PCFF = 000100	LPOV = 000140	LPR = 000300	LSR = 177560
LADR = 000001	L0AD = 007100	LBC = 000002	LBYT = 000000
LCKSH = 000003	LDRWR0 = 007126	LHALT = 007200	LJMP = 007146
LJMP1 = 007202	LLO2 = 007022	LLO3 = 007066	LLO4 = 007110
LPTX = 007114	MAJOR = 006072	MAK3X = 007600	MAK5Y = 000077
MAXX = 001777	MAXY = 001377	MINUY = 000100	MINUSX = 000000
MINUSY = 000000	M7 = 002252	M8 = 000275	M9 = 000301
NORMAL = 006212	NOTHER = 006042	NUMLIN = 000040	NXTGHR = 000132
ORIGIN = 166000	OTHR = 007726	OUTLIT = 000052	O2A = 000146
O2A = 002176	PC = 000007	POINT = 110000	POINTR = 000001
PRESTR = 006624	PRG0 = 001034	PRO0R = 001040	PRO1 = 001006
PRO1A = 001944	PRG1B = 001562	PRO1C = 001000	PRO10 = 001000
PRO2 = 001070	PRMTR8 = 001024	PSW = 177776	PTB00T = 007400
PRWFAL = 000024	P0IB = 002512	P0IC = 000772	P0IS = 002510
P0OP = 002516	P0OC = 004576	P0OS = 000014	RCVEC = 007764
RC11 = 007720	RELATV = 130000	RES = 007900	RESTRY = 000660
RETURN = 002130	RET1 = 000000	RFVEC = 007714	RF11 = 007600
RKVEC = 007760	RK11 = 007610	ROMA00 = 001000	ROMORG = 166000
RPVEC = 007770	RP11 = 007654	RSTRY = 000034	R0 = 000000
R1 = 000001	R2 = 000002	R3 = 000003	R4 = 000004
R5 = 000005	R6 = 000006	R7 = 000007	SCAN = 000003

SCOPE = 174000	SCOPEB = 002132	SCOPEC = 000054	SCOPEF = 002126
SCOPEG = 002112	SCR1 = 000005	SENDIT = 000730	SETOUN = 000120
SETLP1 = 006074	SETLP2 = 006110	SETLP3 = 000110	SETUP = 006432
SWORTV = 164000	SP = 000000	SR = 177570	START = 000000
STARTA = 160000	STARTX = 000000	STARTY = 001360	STATSA = 170000
STATSB = 174000	STKPTR = 000500	STKSRT = 000770	STOP = 007550
SWITCH = 002306	SYNON = 000004	TAB = 000222	TABCNT = 000002
TABLE = 007562	TABOOT = 007500	TACS = 177500	TAPES = 007660
TCVEC = 007774	TC11 = 007620	TERH = 000422	THPEND = 007776
TM11 = 007630	TPCSR = 177564	TPDSR = 177560	TYPM = 001712
TYPMA = 001720	TYPMB = 001750	TYPMC = 000030	T1 = 001660
T1A = 001706	T1B = 001136	T2 = 001140	T2A = 001156
T2B = 001202	T2C = 001214	T2D = 001224	T2E = 001234
T3 = 001244	T3A = 001266	T3AA = 001260	T3B = 001276
T3C = 001304	T3D = 001312	T3E = 001322	T3F = 001332
T4 = 001352	T4A = 001366	T4B = 001402	T4E = 001416
VT = 006244	VT40PC = 002000	WORDS = 000002	WORK1 = 000003
WORK2 = 000004	.		

ERRORS DETECTED: 0

REM *

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DDGTB-B-D
PRODUCT NAME:	GT42/GT44 INSTRUCTION TEST II
DATE CREATED:	NOVEMBER 1, 1973
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	RAYMOND SHOOP

COPYRIGHT (C) 1973, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED TO PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DEC'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DEC.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.

DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.

1. ABSTRACT

THIS IS A TWO PART LOGIC TEST OF THE ALPHAGRAPHIC TERMINAL.
FOR THIS TEST THE TWO MAINTENANCE SWITCH WILL NOT BE USED.
THIS TEST IS DESIGNED TO TEST ALL FUNCTIONAL REGISTERS AND INTERRUPT
VECTOR IN THE ALPHAGRAPHIC DISPLAY CONTROL.
THIS PROGRAM DOES NOT TYPEOUT OR DISPLAY ANY MESSAGES.
THE PROGRAM WILL ONLY HALT ON AN ERROR.

2. REQUIREMENTS

2.1 EQUIPMENT

GT48 DISPLAY SYSTEM (REF. 7.) OR
GT44 DISPLAY SYSTEM

2.2 STORAGE

THIS PROGRAM USED MEMORY LOCATIONS 00140000 <LESS THAN 4K OF MEMORY>.

3. LOADING PROCEDURE

3.1 METHOD

PROCEDURE FOR NORMAL BINARY TAPES SHOULD BE FOLLOWED.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

SWITCH BIT 14 = 1 LOOP ON TEST

4.2 STARTING ADDRESS OR ADDRESSES

200 SUB-TEST 1, COMPLEX LOGIC TEST <DR, NPR AND INTERRUPT>
204 SUB-TEST 2, BASIC VISUAL DISPLAY PATTERNS
 <SELECTED BY SW 00002>

0	POSITIVE HORIZONTAL LINE FROM CENTER SCREEN
1	NEGATIVE HORIZONTAL LINE FROM CENTER SCREEN
2	POSITIVE VERTICAL LINE FROM CENTER SCREEN
3	NEGATIVE VERTICAL LINE FROM CENTER SCREEN
4	RECTANGLE AROUND SCREEN EDGE
5	OCTAGON PATTERN IN RELATIVE POINT AND SHORT VECTOR
6	CHARACTER SET
7	LIGHT PEN TEST

5. OPERATING PROCEDURE

NONE: ONCE STARTED BOTH SUB-TESTS WILL RUN IN THEIR NORMAL MANNER WITHOUT OPERATOR INTERVENTION OR SWITCH SELECTION.

6. ERRORS

THE PROGRAM WILL ONLY HALT ON AN ERROR.
THE PROGRAM DOES NOT CONTAIN FACILITIES FOR REPORTING MESSAGES OR ERROR CONDITIONS. TO PLACE THE PROGRAM INTO A SCOPE LOOP, REPLACE THE ERROR HALT WITH A NOP, SET SWITCH 14 = 1 AND DEPRESS CONT.

7. RESTRICTIONS

BOTH SUB-TESTS DO NOT USE THE MAINTENANCE SWITCHES.
IF VR14 SCOPE, LOCATION "GSYAXIS" (LOC. 1012) MUST BE CHANGED TO 1377.

8. MISCELLANEOUS

8.1 EXECUTION TIME

SUB-TEST 1 TAKES APPROXIMATELY 35 SECONDS,
N/A OPERATOR INTERVENTION ONLY.

8.2 DEVICE ADDRESS PROGRAM LOCATIONS

LOCATION 1000 CONTAINS THE GT40/GT44 DEVICE ADDRESS
LOCATION 1002 CONTAINS THE GT40/GT44 INTERRUPT VECTOR,
LOCATION 1004 CONTAINS THE GT40/GT44 INTERRUPT LEVEL,
LOCATION 1006 CONTAINS THE GT40/GT44 CHARACTER SIZE,
LOCATION 1010 CONTAINS THE GT40/GT44 LINE FEED SIZE,
LOCATION 1012 CONTAINS THE GT40/GT44 +Y AXIS CUTOFF LOCATION;
(LOC. 1012 = 1377 IF VR14 SCOPE)
(LOC. 1012 = 1777 IF VR17 SCOPE)

9. PROGRAM DESCRIPTION

9.1 SUBTEST 1

THIS SUBTEST IS A COMPLEX TEST OF THE DISPLAY STATUS, X AXIS AND Y AXIS REGISTERS, THE PROGRAM ALSO TESTS STOP<DONE>, LIGHT-PEN, TIME-OUT AND SHIFT-OUT INTERRUPTS AND VECTORS, ALSO INCLUDED ARE TESTS FOR MODE, LINE-TYPE, BLINK, INTENSITY LEVELS, ITALICS AND COLOR CHANGE, THE 'RESUME' <STEP> INSTRUCTION IS USED TO SINGLE STEP THRU THE DISPLAY FILE, ALL DISPLAY INSTRUCTIONS ARE TESTED FOR PROPER OPERATION, TESTS ARE ALSO MADE FOR SETTING OF THE 'EDGE' FLAG, WHEN EXCEEDING ALL FOUR DISPLAY EDGES, TESTS ARE ALSO MADE THAT 'NULL', 'ORI', 'LFI' AND 'BSI' CHANGE X OR Y AXIS CORRECTLY.

9.2 SUBTEST 2

THIS SUBTEST CONSISTS OF SEVERAL BASIC VISUAL DISPLAY PATTERNS TO AID IN THE REPAIR AND ALIGNMENT OF THE GT-40 TERMINAL, ONCE A PATTERN HAS BEEN SELECTED BY SW 00-02, THE PROGRAM MUST BE RESTARTED TO SELECT ANOTHER PATTERN.

```

ENABL ABS:AMA
TITLE GT=48/GT=44 INSTRUCTION TEST II MAINDEC=11=DDGTB=8
LIST ME,WIN,SEQ
NLIST MC,PD,CND

256      001030 000000      100
257      001030 000000      WALT
258      001032 000000      WALT
259      001032 000000      IS THRU 776 IS FILLED WITH A TRAR CATCHER
260      001024 000024      104
261      001024 012364      LOWPWR
262      001026 000340      340
263      001030 000030      1030
264      001032 012320      WORD SCOPEA IEWT RETURN
265      001032 000340      340
266      001030 000200      10200
267      001030 000137 001336      JMP START
268      001034 000137 012464      JMP START1
269      001000 001000      10000
270      001000 172000      DRADD; 172000      100 DISPLAY STARTING ADDRESS
271      001002 000320      DRVCT; 320      100 DISPLAY STARTING VECTOR
272      001004 000200      DRBRI; 200      100 DISPLAY INTERRUPT LEVEL
273      001006 000016      DRHBS; 16      10 CHARACTER SIZE (14=16)
274      001008 000030      DRWPS; 30      10 LINE FEED SIZE (30=22)
275      001012 001777      DRVAX; 1777      10 V AXIS CUTOFF LOCATION
276      001014 000177      DRBNO; 177      10 SHIFT=OUT END CHARACTER
277      001016 000000      IDNT; 0      10 PASS COUNTER
278      001000 177776      PSN; 177776      10 FIRST WORD IN THE DISPLAY BUFFER
279      001002 013464      DRBFI; BUFFER+2      10 SECOND WORD
280      001004 013464      DRBFI; BUFFER+4      10 THIRD WORD
281      001006 013472      DRBFI; BUFFER+6      10 FOURTH WORD
282      001008 013474      DRBFI; BUFFER+8      10 FIFTH WORD
283      001010 013476      DRBFI; BUFFER+10      10 SIX WORD
284      001012 000000      DRBFI; BUFFER+12      10 SEVEN WORD
285      001014 177776      SWR; 177776      10 BUFFER SIZE FOR 4K (WORD LENGTH)
286      001016 017476      SIZE; 17476      10 LINE FEED DELTA Y SIZE
287      001018 000000      CNTR; 0      10 BACK SPACE CHARACTER DELTA X SIZE
288      001020 000750      LPSIZE; 750
289      001022 000762      CHSIZE; 762

```

```

323
327
332
333
334 001032 172000      DPC; 172000      10 DISPLAY PD REGISTER
335 001034 172002      DSR; 172002      10 DISPLAY STATUS REGISTER
336 001036 172004      XPSI; 172004      10 X AXIS REGISTER (READ ONLY)
337 001038 172006      YPSI; 172006      10 Y AXIS REGISTER AND GRAPHPLT REGISTER (READ ONLY)
338
339 001002 000320      DDONE; 320      10 DISPLAY STOP (DONE) VECTOR
340 001004 000322      DDONE; 322
341
342 001006 000324      LPVCT; 324      10 DISPLAY LIGHT PEN VECTOR
343 001008 000326      LPVCT; 326
344
345 001072 000330      TIMEVT; 330      10 DISPLAY TIME=OUT (NXN) ERROR VECTOR
346 001074 000332      TIMEVT; 332      10 OR "SHIFT=OUT" VECTOR
347
348
349
350 001076 012700 001032      SETUP; MOV 001032      10 SET UP POINTER
351 001102 013701 001000      MOV 001000      10
352 001106 010120      SETUP; MOV R1,0
353 001110 002701 000002      ADD 002701      10
354 001114 022700 001062      DMP 001062      10
355 001120 001372      BNE 001372      10
356 001122 012700 001062      MOV 001062      10
357 001126 013701 001002      MOV 001002      10
358 001132 010120      SETUP; MOV R1,0
359 001134 002701 000002      ADD 002701      10
360 001136 022700 001076      DMP 001076      10
361 001144 001372      BNE 001372      10
362 001146 013737 001010 001046      MOV 001046      10
363 001104 005437 001046      NEG 001046      10
364 001100 042737 177000 001046      SID 177000      10
365 001106 013737 001006 001050      MOV 001006      10
366 001174 005437 001050      NEG 001050      10
367 001200 004737 001252      JSR 001252      10
368 001204 042737 177000 001050      SIC 177000      10
369 001212 013777 001044 177642      MOV 001044      10
370 001220 005077 177640      CLR 005077      10
371 001224 013777 001076 177634      MOV 001076      10
372 001232 005077 177632      CLR 005077      10
373 001236 013777 001074 177626      MOV 001074      10
374 001244 005077 177624      CLR 005077      10
375 001250 000207      RTS 000207      10

```

```

377      ;SUBROUTINE TO DETERMINE THE SIZE OF CORE
378      ; AND SET UP LOCATION SIZE WITH THE VALUE
379
380      001252 012737 001304 000004 DD CORE: MOV 000004      ;SET UP FOR NEW
381      001260 012701 017776      MOV 017776,R1      ;SET UP ADDRESS
382      001264 000000      CLR R0
383      001266 062701 020000      1ST AOD 000000,R1      ;MOVE TO THE NEXT BANK
384      001272 000200      INC R0      ;INC BANK COUNTER
385      001274 000011      TST (R1)      ;TIMEOUT ?
386      001276 022701 177776      CMP 017776,R1      ;END ?
387      001302 001371      BNE 10
388      001304 000300      DEC R0      ;DECREMENT BANK COUNT
389      001306 012737 000004 000004 2ST MOV 000004      ;RESET SUBS ERROR
390      001314 022626      CMP (R0), (R0)      ;POP THE STACK R0
391      001316 162701 020000      SUB 000000,R1
392      001322 001337 001042      MOV R1, R2C      ;LOAD R2C
393      001326 162737 007776 001042      SUB 077776, R2C      ;BYPASS LOADER
394      001304 000207      RTS PC      ;EXIT
395

```

```

397
398      001336 012777 000340 177454 3TART: MOV 034000,R0      ;SET UP POINTER
399      001344 010704 000500      MOV 000500,R0      ;SET UP POINTER
400      001350 004737 001076      JBR PC, RETUR      ;RETURN
401      001354 000037 001016      CLR 10H
402      001360 012701 001366      MOV 001366,R1      ;SET UP SUBS R1
403
404      ;TEST FOR SUBS ERRORS ON DISPLAY ADDRESSES
405
406      001364 104000      QTSUB: SCOPE
407      001366 000000      RESET
408      001370 000077 177460      CLR 0000      ; ON DISPLAY STATUS
409      001374 000240      NOP
410      001376 000077 177454      CLR 0000      ; ON DISPLAY X REGISTER
411      001402 000240      NOP
412      001404 000077 177450      CLR 0000      ; ON DISPLAY Y REGISTERS
413      001410 000000      RESET
414
415      ;INCREMENT P.C. TEST
416      ;COMPLEX - BUFFER LENGTH
417
418      001432 104000      3TRC: SCOPE
419      001434 013702 001022      MOV 000000      ;SET UP POINTER
420      001436 012722 177000      MOV 012722,R1      ;MOVE STOP INTO THE BUFFER
421      001440 023700 001042      CMP 012722,R1      ;FINISHED FILLING THE BUFFER?
422      001430 001373      BNE 10      ;NO
423
424      001432 104000      3DOPE: SCOPE
425      001434 013777 001022 177410      MOV 000000      ;YES, START THE DISPLAY
426      001442 013737 001022 001036      MOV 000000      ;SETUP A COUNT
427      001430 013702 001042      MOV 013702,R2
428      001434 000300      DEC R2
429      001436 017704 177372      3TPCA: MOV 000000,R4
430      001402 100400      BHI 10      ;ERROR, STOP FLAG FAILED TO SET
431      001404 000000      HALT
432      001406 000421      BR 0TS
433
434      001470 062737 000000 001036 3BT AOD 000000      ;READ DISPLAY R.C.
435      001476 017700 177350      MOV 000000      ;C/O IT INCREMENT BY 2?
436      001502 023700 001036      CMP 000000      ;YES
437      001506 001402      SEC 25      ;DISPLAY PD FAILED TO INCREMENT
438      001510 000000      HALT      ;PROPERLY
439      001512 000407      BR 0TS
440
441      001514 020037 001036      2BT CMP 000000      ;FINISHED THE BUFFER ?
442      001506 001404      SEC 0TS      ;OR IF YES
443      001502 012777 000001 177322      MOV 010000      ;SINGLE STEP THE DISPLAY
444      001510 000752      BR 0TPEA      ;TRY AGAIN
445

```

TESTED BY "LOAD CHARACTER" MODE

MACY11,624 25-NOV-73 10158 PAGE 10

```

498
499 001770 104000 C751 SCOPE
500 001772 012777 172000 177022 MOV #170000,0BUF ;COLOR ENABLE=0 COLOR=0
501 002000 013777 001002 177044 MOV 0BUF,00PC ;LOAD DISPLAY P.C.
502 002000 017700 177042 MOV 00R,R0 ;READ DISPLAY STATUS REGISTER
503 002012 042700 177773 BIC #177773,R0 ;MASK TO BIT 2
504 002016 022700 000004 CMP #4,R0 ;TEST R0
505 002022 001401 BEQ ;
506 002024 000240 NOP ;COLOR ENABLE FAILED TO INHIBIT
;RESETTING OF COLOR BIT
507
508
509
510
511 002026 104000
512 002030 012777 100004 176704 C761 SCOPE
513 002036 012777 172000 176700 MOV #100004,0BUF ;LOAD LINE TYPE ENABLE #1 AND LINE TYPE VALUE #0
514 002044 013777 001022 177000 MOV #170000,0BUF ;LOAD STOP
515 002052 017700 176776 MOV 00R,R0 ;LOAD DISPLAY P.C.
516 002056 042700 177774 BIC #177774,R0 ;READ DISPLAY STATUS REGISTER
517 002062 022700 000000 CMP #0,R0 ;MASK TO BITS 1=0
518 002066 001401 BEQ ;TEST R0
519 002070 000000 HALT ;LINE BITS 1=0 FAILED TO RESET
520
521 002072 104000
522 002074 012777 100007 176720 C771 SCOPE
523 002102 012777 172000 176714 MOV #100007,0BUF ;LINE TYPE ENABLE #1 LINE TYPE #3
524 002110 013777 001022 176734 MOV 0BUF,00PC ;LOAD STOP
525 00211A 017700 176732 MOV 00R,R0 ;LOAD DISPLAY P.C.
526 002122 042700 177774 BIC #177774,R0 ;READ DISPLAY STATUS REGISTER
527 002126 022700 000003 CMP #3,R0 ;MASK TO BITS 1=0
528 002132 001401 BEQ ;TEST R0
529 002134 000000 HALT ;LINE BITS 1=0 FAILED TO SET
530
531 002136 104000
532 002140 012777 100005 176634 C781 SCOPE
533 002146 012777 172000 176600 MOV #100005,0BUF ;LINE TYPE ENABLE #1 LINE TYPE #1
534 002154 013777 001022 176670 MOV 0BUF,00PC ;LOAD STOP
535 002162 017700 176666 MOV 00R,R0 ;LOAD DISPLAY P.C.
536 002166 042700 177774 BIC #177774,R0 ;READ DISPLAY STATUS REGISTER
537 002172 022700 000001 CMP #1,R0 ;MASK TO BITS 1=0
538 002176 001401 BEQ ;TEST R0
539 002200 000000 HALT ;LINE BIT 0 FAILED TO SET
540
541
542 002202 104000
543 002204 012777 100006 176610 C791 SCOPE
544 002212 012777 172000 176604 MOV #100006,0BUF ;LINE TYPE ENABLE #1 LINE TYPE #2
545 002220 013777 001022 176624 MOV 0BUF,00PC ;LOAD STOP
546 002226 017700 176622 MOV 00R,R0 ;LOAD DISPLAY P.C.
547 002232 042700 177774 BIC #177774,R0 ;READ DISPLAY STATUS REGISTER
548 002236 022700 000002 CMP #2,R0 ;MASK TO BITS 1=0
549 002242 001401 BEQ ;TEST R0
550 002244 000000 HALT ;LINE BIT 1 FAILED TO SET

```

552	022246	184888		GT801	SCOPE		
553	022250	012777	188883	176544	MOV	#188883,080UF	;LINE TYPE ENABLE #0 LINE TYPE #0
554	022256	012777	172888	176548	MOV	#172888,080UF,1	
555	022244	013777	081822	17A568	MOV	080UF,00PC	;LOAD DISPLAY P.C.
556	022272	017788	176556		MOV	008R,R8	;READ DISPLAY STATUS REGISTER
557	022276	042788	177774		BIC	#177774,R8	;MASK TO BIT 1=0
558	022302	022788	088882		CHP	#2,R8	;TEST R8
559	022306	088141			BEQ		;;SHOULD NOT CHANGE LT VALUE
560	022310	088888			HALT		;LINE TYPE ENABLE FAILED TO INHIBIT
561							CHANGING OF LINETYPE VALUE
562							
563							
564	081312	184888		GT811	SCOPE		
565	022314	012777	188888	176588	MOV	#188888,080UF	;BLINK ENABLE #1 BLINK #0
566	022322	012777	172888	17A474	MOV	#172888,080UF,1	
567	022330	013777	081822	176514	MOV	080UF,00PC	;LOAD DISPLAY P.C.
568	022336	017788	176512		MOV	008R,R8	;READ DISPLAY STATUS REGISTER
569	022342	042788	177767		BIC	#177767,R8	;MASK TO BIT 3
570	022346	022788	088888		CHP	#8,R8	;TEST R8
571	022352	081481			BEQ		;;
572	022354	088888			HALT		;BLINK BIT FAILED TO RESET
573							
574							
575	081356	184888		GT821	SCOPE		
576	022360	012777	188838	176434	MOV	#188838,080UF	;BLINK ENABLE #1 BLINK #1
577	022366	012777	172888	176438	MOV	#172888,080UF,1	
578	022374	013777	081822	176458	MOV	080UF,00PC	;LOAD DISPLAY P.C.
579	022402	017788	17A446		MOV	008R,R8	;READ DISPLAY STATUS REGISTER
580	022406	042788	177767		BIC	#177767,R8	;MASK TO BIT 3
581	022412	022788	088818		CHP	#18,R8	;TEST R8
582	022416	081481			BEQ		;;
583	022428	088888			HALT		;BLINK BIT FAILED TO SET
584							
585							
586	081482	184888		GT131	SCOPE		
587	022484	012777	188888	176378	MOV	#188888,080UF	;BLINK ENABLE #0 BLINK #0
588	022492	012777	172888	176384	MOV	#172888,080UF,1	
589	022496	013777	081822	17A484	MOV	080UF,00PC	;LOAD DISPLAY P.C.
590	022448	017788	17A482		MOV	008R,R8	;READ DISPLAY STATUS REGISTER
591	022492	042788	177767		BIC	#177767,R8	;MASK TO BIT 3
592	022456	022788	088818		CHP	#18,R8	;TEST R8
593	022482	081481			BEQ		;;
594	022484	088888			HALT		;BLINK ENABLE FAILED TO INHIBIT
595							CHANGING OF THE BLINK BIT
596							
597	081466	184888		GT241	SCOPE		
598	022478	012777	188188	176384	MOV	#188188,080UF	;LP ENABLE #1 LP#0
599	022476	012777	172888	176388	MOV	#172888,080UF,1	
600	022584	013777	081822	176348	MOV	080UF,00PC	;LOAD DISPLAY P.C.
601	022512	017788	176336		MOV	008R,R8	;READ STATUS
602	022516	022788	088828		BIC	#288,R8	
603	022522	081481			BEQ		;;
604	022524	088888			HALT		;LIGHT PEN FLAG SET IN ERROR

627	021526	104000		G151	SCOPE		
628	021530	012777	100140	176244	MOV	#100140,00BUP	ILP ENABLE #1 LPM1
629	021536	012777	172000	176240	MOV	#172000,00BUP1	
610	021544	013777	001022	17A300	MOV	00BUP,00C	LOAD DISPLAY P.C.
611	021532	017700	17A276		MOV	00B0,R0	READ STATUS
612	021536	032700	000200		BIF	#200,R0	
613	021562	001401			EQ	,+4	
614	021564	000000			HALT		LIGHT PEN FLAG SET IN ERROR
615							
616	021566	104000		G161	SCOPE		
617	021570	012777	102000	176224	MOV	#102000,00BUP	INTENSITY LEVEL ENABLE #1 LEVEL #0
618	021576	012777	172000	176200	MOV	#172000,00BUP1	
619	021604	013777	001022	176240	MOV	00BUP,00C	LOAD DISPLAY P.C.
620	021612	017700	176236		MOV	00B0,R0	READ DISPLAY STATUS REGISTER
621	021616	042700	174377		BIC	#174377,R0	MASK TO BITS 0-10
622	021602	022700	000000		CHP	#0,R0	TEST R0
623	021606	001401			EQ	,+4	
624	021630	000000			HALT		INTENSITY LEVEL BITS 6-10 FAILED TO RESET
625							
626							
627	021632	104000		G171	SCOPE		
628	021634	012777	103600	176160	MOV	#103600,00BUP	INTENSITY LEVEL ENABLE #1 LEVEL #7
629	021642	012777	172000	17A154	MOV	#172000,00BUP1	
630	021630	013777	001022	176174	MOV	00BUP,00C	LOAD DISPLAY P.C.
631	021636	017700	176172		MOV	00B0,R0	READ DISPLAY STATUS REGISTER
632	021662	042700	174377		BIC	#174377,R0	MASK TO BITS 0-10
633	021666	022700	003400		CHP	#340,R0	TEST R0
634	021672	001401			EQ	,+4	
635	021674	000000			HALT		INTENSITY LEVEL BITS 6-10 FAILED TO SET
636							
637							
638	021676	104000		G181	SCOPE		
639	021700	012777	100000	176114	MOV	#100000,00BUP	INTENSITY LEVEL ENABLE #1 LEVEL #4
640	021706	012777	172000	176110	MOV	#172000,00BUP1	
641	021714	013777	001022	176130	MOV	00BUP,00C	LOAD DISPLAY P.C.
642	021702	017700	17A124		MOV	00B0,R0	READ DISPLAY STATUS REGISTER
643	021726	042700	174377		BIC	#174377,R0	MASK TO BITS 0-10
644	021732	022700	000000		CHP	#200,R0	TEST R0
645	021736	001401			EQ	,+4	
646	022740	000000			HALT		INTENSITY LEVEL BIT 10 FAILED
647							
648							
649	021742	104000		G191	SCOPE		
650	021744	012777	102400	176050	MOV	#102400,00BUP	INTENSITY LEVEL ENABLE #1 LEVEL #2
651	021752	012777	172000	176044	MOV	#172000,00BUP1	
652	021760	013777	001022	176064	MOV	00BUP,00C	LOAD DISPLAY P.C.
653	021766	017700	17A062		MOV	00B0,R0	READ DISPLAY STATUS REGISTER
654	021772	042700	174377		BIC	#174377,R0	MASK TO BITS 0-10
655	021776	022700	001000		CHP	#100,R0	TEST R0
656	021802	001401			EQ	,+4	
657	021804	000000			HALT		INTENSITY LEVEL BIT 9 FAILED

```
659      000076 104000      07201  SCOPE
660      000010 012777 102200 176004      MOV      010000,000UF ;INTENSITY LEVEL ENABLE =1 LEVEL =1
661      000016 012777 170000 176000      MOV      017000,000UF1
662      000024 013777 001022 176020      MOV      000UF,00PC ;LOAD DISPLAY P.C.
663      000032 017700 176016      MOV      000UF,R0 ;READ DISPLAY STATUS REGISTER
664      000036 042700 174377      BIC      0174377,R0 ;MASK TO BITS 0-10
665      000042 022700 000400      CMP      0400,R0 ;TEST R0
666      000046 001401      BEQ      044 ;INTENSITY LEVEL BIT 0 FAILED
667      000050 000000      MALT
668
669
670      000052 104000      07211  SCOPE
671      000094 012777 101600 175740      MOV      010100,000UF ;INTENSITY LEVEL ENABLE =0 LEVEL =7
672      000096 012777 170000 170734      MOV      017000,000UF1
673      000100 013777 001022 170754      MOV      000UF,00PC ;LOAD DISPLAY P.C.
674      000104 017700 175752      MOV      000UF,R0 ;READ DISPLAY STATUS REGISTER
675      000108 042700 174377      BIC      0174377,R0 ;MASK TO BITS 0-10
676      000112 022700 000400      CMP      0400,R0 ;TEST R0
677      000116 001401      BEQ      044 ;INTENSITY LEVEL ENABLE FAILED TO INHIBIT
678      000120 000000      MALT ;INTENSITY LEVEL CHANGE
679
680
681
682      ;GRAPHPLOT INCREMENT REGISTER TEST
683
684      000116 104000      07221  SCOPE
685      000120 012777 174100 175674      MOV      0174100,000UF ;LOAD GRAPHPLOT COUNTER
686      000124 012777 170000 170670      MOV      017000,000UF1
687      000128 013777 001022 170710      MOV      000UF,00PC ;START DISPLAY
688      000132 017700 175710      MOV      000UF,R0 ;READ INCREMENT REGISTER
689      000136 042700 001777      BIC      01777,R0 ;MASK TO BITS 10-18
690      000140 022700 000000      CMP      00,R0
691      000144 001401      BEQ      044 ;GRAPHPLOT REGISTER IN ERROR
692      000148 000000      MALT
693
694      000162 104000      07231  SCOPE
695      000164 012777 174177 170630      MOV      0174177,000UF ;LOAD GRAPHPLOT COUNTER
696      000168 012777 170000 170604      MOV      017000,000UF1
697      000172 013777 001022 170644      MOV      000UF,00PC ;START DISPLAY
698      000176 017700 170644      MOV      000UF,R0 ;READ INCREMENT REGISTER
699      000180 042700 001777      BIC      01777,R0 ;MASK TO BITS 10-18
700      000184 022700 170000      CMP      00,R0
701      000188 001401      BEQ      044 ;GRAPHPLOT REGISTER IN ERROR
702      000192 000000      MALT
703
704      000226 104000      07241  SCOPE
705      000228 012777 174152 170564      MOV      0174150,000UF ;LOAD GRAPHPLOT COUNTER
706      000232 012777 017200 170540      MOV      017000,000UF1
707      000236 013777 001022 170580      MOV      000UF,00PC ;START DISPLAY
708      000240 017700 170580      MOV      000UF,R0 ;READ INCREMENT REGISTER
709      000244 042700 001777      BIC      01777,R0 ;MASK TO BITS 10-18
710      000248 022700 124000      CMP      00,R0
711      000252 001401      BEQ      044 ;GRAPHPLOT REGISTER IN ERROR
712      000256 000000      MALT
```

```
714      000272 104000      07201  SCOPE
715      000274 012777 174120 170500      MOV      0174120,000UF ;LOAD GRAPHPLOT COUNTER
716      000278 012777 170000 170514      MOV      017000,000UF1
717      000282 013777 001022 170534      MOV      000UF,00PC ;START DISPLAY
718      000286 017700 170534      MOV      000UF,R0 ;READ INCREMENT REGISTER
719      000290 042700 001777      BIC      01777,R0 ;MASK TO BITS 10-18
720      000294 022700 000000      CMP      00,R0
721      000298 001401      BEQ      044 ;GRAPHPLOT REGISTER IN ERROR
722      000302 000000      MALT
723
724      000336 104000      07261  SCOPE
725      000338 012777 174100 170404      MOV      0174100,000UF ;LOAD GRAPHPLOT COUNTER WITH 0
726      000342 012777 170000 170400      MOV      017000,000UF1
727      000346 013777 001022 170470      MOV      000UF,00PC ;START DISPLAY
728      000350 004737 010340      JSR      7,0LAY ;EXECUTE A PROGRAM DELAY
729      000354 012777 170077 170404      MOV      0170077,000UF ;LOAD GRAPHPLOT NO ENABLE
730      000358 013777 001022 170400      MOV      000UF,00PC ;START DISPLAY
731      000362 017700 170400      MOV      000UF,R0 ;READ INCREMENT REGISTER
732      000366 042700 001777      BIC      01777,R0 ;MASK TO BITS 10-18
733      000370 022700 000000      CMP      00,R0 ;ARE THEY EQUAL ?
734      000374 001401      BEQ      044 ;GRAPHPLOT REGISTER CHANGED WITHOUT
735      000378 000000      MALT ; THE ENABLE BEING SET
736
737
738      ;TEST THAT THE X POSITION REGISTER CAN BE LOADED CORRECTLY
739      ;USING GRAPHPLOT X
740
741      000422 104000      07271  SCOPE
742      000424 012777 122000 170370      MOV      012000,000UF ;LOW INTENSITY - SET GRAPHPLOT X MODE
743      000428 012777 001252 170304      MOV      012000,000UF1 ;SET X POSITION
744      000432 012777 170000 170300      MOV      017000,000UF2 ;LOAD STOP
745      000436 013777 001022 170376      MOV      000UF,00PC ;START THE DISPLAY
746      000440 004737 010340      JSR      7,0LAY ;EXECUTE A PROGRAM DELAY
747      000444 017700 170372      MOV      000UF,R0 ;READ X POSITION
748      000448 022700 001250      CMP      01000,R0
749      000452 001401      BEQ      044 ;X POSITION REGISTER FAILED TO LOAD
750      000456 000000      MALT ; PROPERLY USING GRAPHPLOT X MODE
751
752
```

```

754
755
756
757
758 000474 104000 122000 175316 07281 SCOPE
759 000476 012777 000525 175312 MOV #12000,00BUF ;LOW INTENSITY - SET GRAPH PLOT X MODE
760 000504 012777 000525 175312 MOV #500,00BUF1 ;SET X POSITION
761 000532 012777 172000 175300 MOV #17000,00BUF2 ;LOAD STATUS REGISTER A: STOP
762 000520 013777 001022 175324 MOV 00UF,00PC ;LOAD THE DISPLAY P.C.
763 000526 004737 012340 JSR 7,0LAY ;EXECUTE A PROGRAM DELAY
764 000532 017700 175320 MOV #XPOS,R0 ;READ X POSITION
765 000536 022700 000525 CMP #000,R0 ;
766 000542 001401 BEQ .+4 ;
767 000544 000000 HALT ;X POSITION REGISTER FAILED TO LOAD
;PROPERLY USING GRAPH PLOT X MODE

768
769
770
771
772
773 000546 104000 126000 170244 07291 SCOPE
774 000548 012777 126000 170240 MOV #10000,00BUF ;LOW INTENSITY - SET GRAPH PLOT Y
775 000556 012777 001252 175040 MOV #500,00BUF1 ;SET Y POSITION
776 000564 012777 172000 175234 MOV #17000,00BUF2 ;LOAD STATUS REGISTER A: STOP
777 000552 013777 001022 175252 MOV 00UF,00PC ;LOAD THE DISPLAY P.C.
778 000600 004737 012340 JSR 7,0LAY ;EXECUTE A PROGRAM DELAY
779 000604 017700 175250 MOV #YPOS,R0 ;READ Y POSITION
780 000608 022700 001252 CMP #100,R0 ;
781 000614 001401 BEQ .+4 ;
782 000616 000000 HALT ;Y POSITION REGISTER FAILED TO LOAD
;PROPERLY USING GRAPH PLOT Y MODE

783
784

```

```

786
787
788
789
790 000620 104000 126000 175172 07301 SCOPE
791 000622 012777 126000 175172 MOV #10000,00BUF ;LOW INTENSITY - SET GRAPH PLOT Y MODE
792 000630 012777 000525 175106 MOV #500,00BUF1 ;SET Y POSITION
793 000636 012777 172000 175102 MOV #17000,00BUF2 ;LOAD STATUS REGISTER A: STOP
794 000644 013777 001022 175200 MOV 00UF,00PC ;LOAD THE DISPLAY P.C.
795 000652 004737 012340 JSR 7,0LAY ;EXECUTE A PROGRAM DELAY
796 000656 017700 175176 MOV #YPOS,R0 ;READ Y POSITION
797 000662 022700 000525 CMP #000,R0 ;
798 000666 001401 BEQ .+4 ;
799 000670 000000 HALT ;Y POSITION REGISTER FAILED TO LOAD
;PROPERLY USING GRAPH PLOT Y MODE

800
801
802
803
804
805
806 000672 104000 122000 175120 07311 SCOPE
807 000674 012777 122000 175120 MOV #10000,00BUF ;LOW INTENSITY - SET GRAPH PLOT X MODE
808 000702 012777 001234 175114 MOV #500,00BUF1 ;SET X POSITION
809 000710 012777 126000 175110 MOV #10000,00BUF2 ;SET GRAPH PLOT Y MODE
810 000716 012777 001432 175104 MOV #1400,00BUF3 ;SET Y POSITION
811 000724 012777 172000 175100 MOV #17000,00BUF4 ;LOAD STATUS REGISTER A: STOP
812 000732 013777 001022 175112 MOV 00UF,00PC ;LOAD THE DISPLAY P.C.
813 000740 004737 012340 JSR 7,0LAY ;EXECUTE A PROGRAM DELAY
814 000744 017700 175106 MOV #XPOS,R0 ;READ X POSITION
815 000750 022700 001234 CMP #100,R0 ;
816 000754 001402 BEQ .+6 ;
817 000756 000000 HALT ;GRAPH PLOT X MODE FAILED TO SELECT
;X POSITION PROPERLY
818 000760 000406 BR 0730
819
820 000762 017700 170272 MOV #YPOS,R0 ;READ Y POSITION
821 000766 022700 001432 CMP #1400,R0 ;
822 000772 001401 BEQ .+4 ;
823 000774 000000 HALT ;Y POSITION REGISTER FAILED TO LOAD
;PROPERLY USING GRAPH PLOT Y MODE

824
825

```

```

827
828
829
830
831 002776 104000 116000 17501A 01321 SCOPE
832 004000 012777 116000 17501A MOV #110000,00BUF1 ;LOW INTENSITY = POINT MODE
833 004000 005077 175012 CLR #00BF1 ;CLEAR X POSITION
834 004012 005077 175010 CLR #00BF2 ;CLEAR Y POSITION
835 004016 012777 175000 17500A MOV #175000,00BUF3 ;LOAD STATUS "A" REGISTER, STOP
836 004024 013777 001022 175020 00BF1,00FC ;LOAD DISPLAY R,C,
837 004032 004737 012340 JSR 7,0LAY ;EXECUTE A PROGRAM DELAY
838 004036 017700 17501A MOV #XPOS,R0 ;READ X POSITION
839 004042 001402 SEQ #+6 ;WAS IT SET?
840 004044 000000 HALT ;X POSITION REGISTER FAILED TO RESET
841 004046 000000 BR GT33 ;USING POINT DATA MODE
842
843 004050 017700 17500A MOV #YPOS,R0 ;READ Y POSITION
844 004054 001401 SEQ #+4 ;WAS IT SET?
845 004056 000000 HALT ;Y POSITION REGISTER FAILED TO RESET
846 ;USING POINT DATA MODE
847
848
849
850
851 004060 104000 116000 17A732 01331 SCOPE
852 004062 012777 116000 17A732 MOV #110000,00BUF1 ;LOW INTENSITY = POINT MODE
853 004070 012777 001777 17A726 MOV #1777,00BUF1 ;SET X POSITION
854 004076 012777 001777 17A722 MOV #1777,00BUF2 ;SET Y POSITION
855 00410A 012777 175000 17A716 MOV #175000,00BUF3 ;LOAD STATUS A REGISTER, STOP
856 004112 013777 001022 17A732 00BF1,00FC ;LOAD DISPLAY R,C,
857 004120 004737 012340 JSR 7,0LAY ;EXECUTE A PROGRAM DELAY
858 004124 017700 17A726 MOV #XPOS,R0 ;READ X POSITION
859 004130 022700 001777 CMP #1777,R0 ;WAS IT SET?
860 004134 001A02 SEQ #+6 ;X POSITION REGISTER FAILED TO SET
861 004136 000000 HALT ;USING POINT DATA MODE
862 004140 000A06 BR GT33
863
864 004142 017700 17A712 MOV #YPOS,R0 ;READ Y POSITION
865 004146 022700 001777 CMP #1777,R0 ;WAS IT SET?
866 004152 001401 SEQ #+4 ;Y POSITION REGISTER FAILED TO SET
867 004154 000000 HALT ;USING POINT DATA MODE
868
869

```

```

871
872
873
874
875 004156 104000 116000 17A63A 01341 SCOPE
876 004158 012777 116000 17A63A MOV #110000,00BUF1 ;LOW INTENSITY = POINT MODE
877 004166 012777 001252 17A630 MOV #1252,00BUF1 ;SET X POSITION
878 00417A 012777 001252 17A624 MOV #1252,00BUF2 ;SET Y POSITION
879 004202 012777 175000 17A620 MOV #175000,00BUF3 ;LOAD STATUS REGISTER A, STOP
880 004210 013777 001022 17A63A MOV 00BF1,00FC ;LOAD STATUS REGISTER A, STOP
881 004216 004737 012340 JSR 7,0LAY ;EXECUTE A PROGRAM DELAY
882 004222 017700 17A630 MOV #XPOS,R0 ;READ X POSITION
883 004226 022700 001252 CMP #1252,R0 ;WAS IT SET?
884 004232 001402 SEQ #+6 ;X POSITION REGISTER FAILED
885 004234 000000 HALT ;USING POINT DATA MODE
886 004236 000A06 BR GT33
887
888 004240 017700 17A614 MOV #YPOS,R0 ;READ Y POSITION
889 00424A 022700 001252 CMP #1252,R0 ;WAS IT SET?
890 004250 001401 SEQ #+4 ;Y POSITION REGISTER FAILED
891 004252 000000 HALT ;USING POINT DATA MODE
892
893
894
895
896
897 004254 104000 116000 17A536 01351 SCOPE
898 004256 012777 116000 17A536 MOV #110000,00BUF1 ;LOW INTENSITY = POINT MODE
899 00426A 012777 000525 17A530 MOV #0525,00BUF1 ;SET X POSITION
900 004272 012777 000525 17A526 MOV #0525,00BUF2 ;SET Y POSITION
901 004300 012777 175000 17A522 MOV #175000,00BUF3 ;LOAD STATUS REGISTER A, STOP
902 004306 013777 001022 17A536 MOV 00BF1,00FC ;LOAD STATUS REGISTER A, STOP
903 004314 004737 012340 JSR 7,0LAY ;EXECUTE A PROGRAM DELAY
904 004320 017700 17A532 MOV #XPOS,R0 ;READ X POSITION
905 004324 022700 000525 CMP #0525,R0 ;WAS IT SET?
906 004330 001A02 SEQ #+6 ;X POSITION REGISTER FAILED
907 004332 000000 HALT ;USING POINT DATA MODE
908 004334 000A06 BR GT33
909
910 004336 017700 17A516 MOV #YPOS,R0 ;READ Y POSITION
911 004342 022700 000525 CMP #0525,R0 ;WAS IT SET?
912 004346 001401 SEQ #+4 ;Y POSITION REGISTER FAILED
913 004350 000000 HALT ;USING POINT DATA MODE
914
915

```

GT=42/RY=44 INSYRUCYION YESY II MAINOC0-11400GTB-8
DDGYB87011

MACT111624 25-NOV-73 10158 PAGE 20

941				TEST THAT LONG VECTOR MODE INCREMENTS X AND Y AXIS PROPERLY
942				ICOUNT 1
943				
944				
945	004450	104000	001022	GT371 SCODE
946	004402	013700		MOV
947	004406	012720	110000	MOV
948	004402	005020		CLR
949	004404	005020		CLR
950	004406	012720	110000	MOV
951	004472	012720	000001	MOV
952	004476	012720	000001	MOV
953	004502	012710	172000	MOV
954	004506	013777	001022	MOV
955	004514	004737	010340	JSR
956				
957	004520	017700	174332	MOV
958	004524	022700	000001	CMP
959	004530	001402		SEC
960	004532	000000		MALT
961	004534	000406		BR
962				
963	004536	017700	174316	MOV
964	004542	022700	000001	CMP
965	004546	001401		SEC
966	004550	000000		MALT
967				
968				
969				
970				
971				
972	004502	104000	001022	GT381 SCODE
973	004504	013700		MOV
974	004500	012720	110000	MOV
975	004504	005020		CLR
976	004506	005020		CLR
977	004570	012720	110000	MOV
978	004574	012720	020001	MOV
979	004600	012720	020001	MOV
980	004604	012710	172000	MOV
981	004606	013777	001022	MOV
982	004606	004737	012340	JSR
983				
984	004602	017700	174230	MOV
985	004606	022700	001777	CMP
986	004632	001402		SEC
987	004634	000000		MALT
988	004636	000406		BR
989				
990	004604	017700	174214	MOV
991	004604	022700	001777	CMP
992	004608	001401		SEC
993	004602	000000		MALT
994				

```

996
997
998
999
1000
1001 004634 104000 001777
1002 004636 012703 001777
1003 004662 012704 000001
1004 004664 104000
1005 004670 013700 001022
1006 004674 012720 110000
1007 004700 005020
1008 004702 005020
1009 004704 012720 110000
1010 004706 014020
1011 004722 014020
1012 004734 012720 172000
1013 004736 013777 001022 174124
1014 004738 004737 012340
1015
1016 004732 017700 174120
1017 004736 020400
1018 004740 001402
1019 004742 000000
1020 004744 000411
1021
1022 004746 017700 174106
1023 004752 020400
1024 004754 001402
1025 004756 000000
1026 004760 000403
1027
1028 004762 005204
1029 004764 005303
1030 004766 001340

;TEST THAT LONG VECTOR MODE INCREMENT X AND T AXIS PROPERLY
;COUNT 0=1777
GT391  SCOPE
      MOV  #1777,R3
      MOV  #1,R4
      ;SET UP A COUNTER
      ;PRESET THE COMPARED VALUE

GT39A;  SCOPE
      MOV  0000,R0
      MOV  #110000,(R0)
      CLR  (R0)
      CLR  (R0)
      MOV  #110000,(R0)
      MOV  R4,(R0)
      MOV  #177000,(R0)
      MOV  000000PC
      JSR  7,0LAT
      ;LOAD THE DISPLAY P.C.
      ;EXECUTE A PROGRAM DELAY

      MOV  #XPOS,R0
      CMP  R4,R0
      BEQ  +6
      HALT
      BR   GT40
      ;READ X AXIS
      ;ARE THEY EQUAL?
      ;YES
      ;NO, INCREMENT X AXIS TIA
      ;LONG VECTOR MODE FAILED

      MOV  #YPOS,R0
      CMP  R4,R0
      BEQ  +6
      HALT
      BR   GT40
      ;READ Y AXIS
      ;ARE THEY EQUAL?
      ;YES
      ;NO, INCREMENT Y AXIS TIA
      ;LONG VECTOR MODE FAILED

      INC  R4
      DEC  R3
      BNE  GT39A
      ;INCREMENT EXPECTED VALUE
      ;FINISHED?
      ;NO, TEST MORE DATA

```

```

1032
1033
1034
1035
1036 004770 104000
1037 004772 012703 002000
1038 004776 012704 001777
1039 004802 012700 000001
1040
1041 005006 104000
1042 005010 013700 001022
1043 005014 012720 110000
1044 005020 005020
1045 005022 005020
1046 005024 012720 110000
1047 005026 010000
1048 005032 010000
1049 005034 012720 172000
1050 005040 013777 001022 174004
1051 005046 004737 012340
1052
1053 005052 017700 174000
1054 005056 020400
1055 005060 001402
1056 005062 000000
1057 005064 000412
1058
1059 005066 017700 173766
1060 005072 020400
1061 005074 001402
1062 005076 000000
1063 005100 000404
1064
1065 005102 005200
1066 005104 005304
1067 005106 005303
1068 005110 001337

;TEST THAT LONG VECTOR MODE DECREMENTS X AND T AXIS PROPERLY
;COUNT 1777=0
GT401  SCOPE
      MOV  #2000,R3
      MOV  #1777,R0
      MOV  #2000,R3
      ;SET UP A COUNTER
      ;PRESET THE COMPARED VALUE

GT40A;  SCOPE
      MOV  0000,R0
      MOV  #110000,(R0)
      CLR  (R0)
      CLR  (R0)
      MOV  #110000,(R0)
      MOV  R3,(R0)
      MOV  #177000,(R0)
      MOV  #177000,(R0)
      MOV  000000PC
      JSR  7,0LAT
      ;LOAD THE DISPLAY P.C.
      ;EXECUTE A PROGRAM DELAY

      MOV  #XPOS,R0
      CMP  R4,R0
      BEQ  +6
      HALT
      BR   GT41
      ;READ X AXIS
      ;ARE THEY EQUAL?
      ;YES
      ;NO, DECREMENT X AXIS VIA
      ;LONG VECTOR MODE FAILED

      MOV  #YPOS,R0
      CMP  R4,R0
      BEQ  +6
      HALT
      BR   GT41
      ;READ Y AXIS
      ;ARE THEY EQUAL?
      ;YES
      ;NO, DECREMENT Y AXIS VIA
      ;LONG VECTOR MODE FAILED

      INC  R3
      DEC  R4
      DEC  R3
      BNE  GT40A
      ;INCREMENT "DELTA X=T"
      ;DECREMENT EXPECTED VALUE
      ;FINISHED?
      ;NO, TEST MORE DATA

```

```

1070
1071
1072
1073
1074 005112 104000
1075 005114 013700 001022
1076 005120 012720 110000
1077 005124 005020
1078 005126 005020
1079 005130 012720 104000
1080 005134 012720 000201
1081 005140 012710 172000
1082 005144 013777 001022 173700
1083 005152 004737 012340
1084
1085 005156 017700 173674
1086 005162 022700 000001
1087 005166 001402
1088 005170 000000
1089 005172 000406
1090
1091 005174 017700 173600
1092 005200 022700 000001
1093 005204 001401
1094 005206 000000
1095
1096
1097
1098
1099
1100
1101 005210 104000
1102 005212 013700 001022
1103 005216 012720 110000
1104 005222 005020
1105 005224 005020
1106 005226 012720 104000
1107 005232 012720 000301
1108 005234 012710 172000
1109 005242 013777 001022 173602
1110 005250 004737 012340
1111
1112 005254 017700 173576
1113 005260 022700 001777
1114 005264 001402
1115 005266 000000
1116 005270 000406
1117
1118 005272 017700 173562
1119 005276 022700 001777
1120 005302 001401
1121 005304 000000
1122

TEST THAT X AND Y AXIS INCREMENTS PROPERLY
USING SHORT VECTOR MODE
COUNT 1

GT411 SCOPE
MOV DBUF,R0 ;SET UP R0
MOV #110000,(0)+ ;LOAD "SET POINT MODE"
CLR (0)+ ;CLEAR X AXIS
CLR (0)+ ;CLEAR Y AXIS
MOV #100000,(0)+ ;LOAD "SET SHORT VECTOR MODE"
MOV #201,(0)+ ;PRESET "DELTA X AND DELTA Y"
MOV #170000,(0)
MOV DBUF,OPPC ;LOAD THE DISPLAY PC
JSR 7,DELAY ;EXECUTE A PROGRAM DELAY

MOV #XPOS,R0 ;READ X AXIS
CMP #1,R0 ;ARE THEY EQUAL?
BEQ ,+6 ;YES
HALT ;NO, INCREMENT X AXIS FAILED USING
SR GT43 ;SHORT VECTOR MODE

MOV #YPOS,R0 ;READ Y AXIS
CMP #1,R0 ;ARE THEY EQUAL?
BEQ ,+6 ;YES
HALT ;NO INCREMENT Y AXIS FAILED
USING SHORT VECTOR MODE

TEST THAT X AND Y AXIS DECREMENT PROPERLY
USING SHORT VECTOR MODE
COUNT 1

GT421 SCOPE
MOV DBUF,R0 ;SET UP R0
MOV #110000,(0)+ ;LOAD "SET POINT MODE"
CLR (0)+ ;CLEAR X AXIS
CLR (0)+ ;CLEAR Y AXIS
MOV #100000,(0)+ ;LOAD "SET SHORT VECTOR MODE"
MOV #201,(0)+ ;PRESET "DELTA X AND DELTA Y"
MOV #170000,(0)
MOV DBUF,OPPC ;LOAD THE DISPLAY PC
JSR 7,DELAY ;EXECUTE A PROGRAM DELAY

MOV #XPOS,R0 ;READ X AXIS
CMP #1777,R0 ;ARE THEY EQUAL?
BEQ ,+6 ;YES
HALT ;NO, DECREMENT X AXIS FAILED USING
SR GT43 ;SHORT VECTOR MODE

MOV #YPOS,R0 ;READ Y AXIS
CMP #1777,R0 ;ARE THEY EQUAL?
BEQ ,+6 ;YES
HALT ;NO DECREMENT Y AXIS FAILED
USING SHORT VECTOR MODE

```

```

1124
1125
1126
1127
1128
1129 005306 104000
1130 005310 012700 000077
1131 005314 012702 000001
1132 005316 012704 000201
1133
1134 005324 104000
1135 005326 013700 001022
1136 005332 012700 110000
1137 005336 005020
1138 005340 005020
1139 005342 012700 104000
1140 005346 010400
1141 005350 012710 172000
1142 005354 013777 001022 173470
1143 005362 004737 012340
1144
1145 005366 017700 173404
1146 005372 020200
1147 005374 001402
1148 005376 000000
1149 005400 000413
1150
1151 005402 017700 173452
1152 005406 020200
1153 005410 001402
1154 005412 000000
1155 005414 000405
1156
1157 005416 002704 000201
1158 005422 005202
1159 005424 005373
1160 005426 001337

TEST THAT X AND Y AXIS INCREMENT PROPERLY
USING SHORT VECTOR MODE
COUNT 8=77

GT431 SCOPE
MOV #77,R3 ;SET UP A COUNT LOCATION
MOV #1,R2 ;SET UP THE COMPARED LOCATION
MOV #201,R4 ;SET UP "DELTA X=Y"

GT43A1 SCOPE
MOV DBUF,R0 ;SET UP R0
MOV #110000,(0)+ ;LOAD "SET POINT DATA MODE"
CLR (0)+ ;CLEAR X AXIS
CLR (0)+ ;CLEAR Y AXIS
MOV #100000,(0)+ ;LOAD "SET SHORT VECTOR MODE"
MOV #4,(0)+ ;PRESET "DELTA X AND DELTA Y"
MOV #170000,(0)
MOV DBUF,OPPC ;LOAD THE DISPLAY P.C.
JSR 7,DELAY ;EXECUTE A PROGRAM DELAY

MOV #XPOS,R0 ;READ X POSITION
CMP #2,R0 ;ARE THEY EQUAL?
BEQ ,+6 ;YES
HALT ;INCREMENT X AXIS FAILED USING
SR GT43 ;SHORT VECTOR MODE

MOV #YPOS,R0 ;READ Y POSITION
CMP #2,R0 ;ARE THEY EQUAL?
BEQ ,+6 ;YES
HALT ;INCREMENT Y AXIS FAILED USING
SR GT43 ;SHORT VECTOR MODE

ADD #201,R4 ;ADD DELTA X=Y
INC R2 ;INCREMENT EXPECTED VALUE
DEC R3 ;INCREMENT COUNT, FINISHED?
BNE GT43A ;NO, TEST MORE DATA

```

```

1162
1163
1164
1165
1166
1167 201430 104000
1168 001432 012703 000077
1169 001436 012702 001777
1170 001442 012704 000301
1171
1172 001446 104000
1173 001450 013700 001022
1174 001454 012720 110000
1175 001460 005020
1176 001462 005020
1177 001464 012720 106000
1178 001470 010400
1179 001472 012710 172000
1180 001476 013777 001022 173346
1181 001504 004737 012340
1182
1183 001510 017700 173342
1184 001514 020200
1185 001516 001402
1186 001520 000000
1187 001522 000413
1188
1189 001524 017700 173330
1190 001526 020200
1191 001532 001402
1192 001534 000000
1193 001536 000413
1194
1195 001540 062704 000201
1196 001544 005302
1197 001546 005303
1198 001550 001337
1199

```

TEST THAT X AND Y AXIS DECREMENT PROPERLY
USING SHORT VECTOR MODE
ICOUNT 77=0

GT441 SCODE
MOV #77,R3
MOV #177,R2
MOV #201,R4
;SET UP A COUNT LOCATION
;SET UP THE COMPARED LOCATION
;PRESET THE "DELTA X=Y"

GT44A SCODE
DBUF#R0
MOV #110001(0)
CLR (0)
CLR (0)
MOV #100001(0)
MOV R4,(0)
MOV #170001(0)
DBUF#R0PC
JBR 7,DELAY
;SET UP R0
;LOAD "SET POINT DATA MODE"
;CLEAR X AXIS
;CLEAR Y AXIS
;LOAD "SET SHORT VECTOR MODE"
;PRESET "DELTA X AND DELTA Y"
;LOAD THE DISPLAY P.C.
;EXECUTE A PROGRAM DELAY
;READ X POSITION
;ARE THEY EQUAL?
;YES
;DECREMENT X AXIS FAILED USING
;SHORT VECTOR MODE
;READ Y POSITION
;ARE THEY EQUAL?
;YES
;DECREMENT Y AXIS FAILED USING
;SHORT VECTOR MODE
;ADD "DELTA X=Y"
;DECREMENT EXPECTED VALUE
;DECREMENT COUNT, FINISHED?
;NO: TEST MORE DATA

```

1201
1202
1203
1204
1205 001552 104000
1206 001554 013700 001022
1207 001560 012720 110000
1208 001564 005020
1209 001566 005020
1210 001570 012720 130000
1211 001574 012720 000201
1212 001580 012710 172000
1213 001584 013777 001022 173240
1214 001612 004737 012340
1215
1216 001616 017700 173234
1217 001622 022700 000001
1218 001626 001402
1219 001630 000000
1220 001632 000406
1221
1222 001634 017700 173220
1223 001640 022700 000001
1224 001644 001401
1225 001646 000000
1226
1227
1228
1229
1230
1231
1232 001650 104000
1233 001652 013700 001022
1234 001656 012720 110000
1235 001662 005020
1236 001664 005020
1237 001666 012720 130000
1238 001672 012720 000301
1239 001676 012710 172000
1240 001702 013777 001022 173142
1241 001710 004737 012340
1242
1243 001714 017700 173136
1244 001720 022700 001777
1245 001724 001402
1246 001726 000000
1247 001730 000406
1248
1249 001732 017700 173122
1250 001736 022700 001777
1251 001742 001401
1252 001744 000000
1253

```

TEST THAT X AND Y AXIS INCREMENTS PROPERLY
USING RELATIVE POINT MODE
ICOUNT 1

GT451 SCODE
MOV DBUF#R0
MOV #110001(0)
CLR (0)
CLR (0)
MOV #130001(0)
MOV #201,R4
MOV #170001(0)
DBUF#R0PC
JBR 7,DELAY
;SET UP R0
;LOAD "SET POINT MODE"
;CLEAR X AXIS
;CLEAR Y AXIS
;LOAD "SET RELATIVE POINT MODE"
;PRESET "DELTA X AND DELTA Y"
;LOAD THE DISPLAY PC
;EXECUTE A PROGRAM DELAY
;READ X AXIS
;ARE THEY EQUAL?
;YES
;INDECREMENT X AXIS FAILED USING
;RELATIVE POINT MODE
;READ Y AXIS
;ARE THEY EQUAL?
;YES
;INDECREMENT Y AXIS FAILED
;USING RELATIVE POINT MODE

TEST THAT X AND Y AXIS DECREMENT PROPERLY
USING RELATIVE POINT MODE
ICOUNT 1

GT461 SCODE
MOV DBUF#R0
MOV #110001(0)
CLR (0)
CLR (0)
MOV #130001(0)
MOV #201,R4
MOV #170001(0)
DBUF#R0PC
JBR 7,DELAY
;SET UP R0
;LOAD "SET POINT MODE"
;CLEAR X AXIS
;CLEAR Y AXIS
;LOAD "SET RELATIVE POINT MODE"
;PRESET "DELTA X AND DELTA Y"
;LOAD THE DISPLAY PC
;EXECUTE A PROGRAM DELAY
;READ X AXIS
;ARE THEY EQUAL?
;YES
;DECREMENT X AXIS FAILED USING
;RELATIVE POINT MODE
;READ Y AXIS
;ARE THEY EQUAL?
;YES
;DECREMENT Y AXIS FAILED
;USING RELATIVE POINT MODE

```

1255                                ;TEST THAT X AND Y AXIS INCREMENT PROPERLY
1256                                ;USING RELATIVE POINT MODE
1257                                ;COUNT 8=77
1258
1259
1260      001746  184888  000077  GT471  SCOPE      ;SET UP A COUNT LOCATION
1261      001738  012783  000077  MOV      #77,R3      ;SET UP THE COMPARED LOCATION
1262      001734  012782  000081  MOV      #1,R2      ;SET UP "DELTA X=Y"
1263      001788  012784  000201  MOV      #2817R4
1264
1265      001764  184888  001022  GT47A1 SCOPE      ;SET UP R8
1266      001766  013788  001022  MOV      0BUT,R8      ;LOAD "SET POINT DATA MODE"
1267      001772  012728  116888  MOV      #11888,(8)+  ;CLEAR X AXIS
1268      001776  005828  005828  CLR      (8)+      ;CLEAR Y AXIS
1269      001808  005828  005828  CLR      (8)+      ;LOAD "SET RELATIVE POINT MODE"
1270      001802  012728  138288  MOV      #13888,(8)+  ;PRESET "DELTA X AND DELTA Y"
1271      001806  018428  005828  MOV      R4,(8)+      ;LOAD THE DISPLAY P.C.
1272      001810  012718  172888  MOV      #17288,(8)  ;EXECUTE A PROGRAM DELAY
1273      001814  013777  001022  MOV      0BUT,00PC
1274      001822  084737  012348  JSR      7,0LAY
1275
1276      001826  017788  173824  MOV      0XPOS,R8      ;READ X POSITION
1277      001832  028288  005828  CMP      R2,R8      ;ARE THEY EQUAL
1278      001834  081482  005828  BEQ      1=6      ;YES
1279      001836  000888  000888  HALT      ;INCREMENT X AXIS FAILED USING
1280      001840  008413  008413  BR      0T48      ;RELATIVE POINT MODE
1281
1282      001842  017788  173812  MOV      0YPOS,R8      ;READ Y POSITION
1283      001846  028288  005828  CMP      R2,R8      ;ARE THEY EQUAL ?
1284      001850  081482  005828  BEQ      1=6      ;YES
1285      001852  000888  000888  HALT      ;INCREMENT Y AXIS FAILED USING
1286      001854  008489  008489  BR      0T48      ;RELATIVE POINT MODE
1287
1288      001856  062784  000281  ADD      #281,R4      ;ADD DELTA X=Y
1289      001862  005382  005382  INC      R2      ;INCREMENT EXPECTED VALUE
1290      001864  005383  005383  DEC      R3      ;INCREMENT COUNT, FINISHED?
1291      001866  001337  001337  BNE      0T47A      ;NO, TEST MORE DATA

```

```

1293                                ;TEST THAT X AND Y AXIS DECREMENT PROPERLY
1294                                ;USING RELATIVE POINT MODE
1295                                ;COUNT 77=8
1296
1297
1298      001878  184888  000077  GT481  SCOPE      ;SET UP A COUNT LOCATION
1299      001872  012783  000077  MOV      #77,R3      ;SET UP THE COMPARED LOCATION
1300      001876  012782  000081  MOV      #1,R2      ;PRESET THE "DELTA X=Y"
1301      001882  012784  000201  MOV      #2817R4
1302
1303      001886  184888  001022  GT48A1 SCOPE      ;SET UP R8
1304      001888  013788  001022  MOV      0BUT,R8      ;LOAD "SET POINT DATA MODE"
1305      001894  012728  116888  MOV      #11888,(8)+  ;CLEAR X AXIS
1306      001898  005828  005828  CLR      (8)+      ;CLEAR Y AXIS
1307      001902  005828  005828  CLR      (8)+      ;LOAD "SET RELATIVE POINT MODE"
1308      001906  012728  138288  MOV      #13888,(8)+  ;PRESET "DELTA X AND DELTA Y"
1309      001910  018428  005828  MOV      R4,(8)+      ;LOAD THE DISPLAY P.C.
1310      001912  012718  172888  MOV      #17288,(8)  ;EXECUTE A PROGRAM DELAY
1311      001916  013777  001022  MOV      0BUT,00PC
1312      001922  084737  012348  JSR      7,0LAY
1313
1314      001926  017788  172782  MOV      0XPOS,R8      ;READ X POSITION
1315      001934  028288  005828  CMP      R2,R8      ;ARE THEY EQUAL
1316      001936  081482  005828  BEQ      1=6      ;YES
1317      001940  000888  000888  HALT      ;DECREMENT X AXIS FAILED USING
1318      001942  008413  008413  BR      0T48      ;RELATIVE POINT MODE
1319
1320      001944  017788  172678  MOV      0YPOS,R8      ;READ Y POSITION
1321      001948  028288  005828  CMP      R2,R8      ;ARE THEY EQUAL ?
1322      001952  081482  005828  BEQ      1=6      ;YES DECREMENT
1323      001954  000888  000888  HALT      ;DECREMENT Y AXIS FAILED USING
1324      001956  008489  008489  BR      0T48      ;RELATIVE POINT MODE
1325
1326      001958  062784  000281  ADD      #281,R4      ;ADD "DELTA X=Y"
1327      001964  005382  005382  DEC      R2      ;DECREMENT EXPECTED VALUE
1328      001966  005383  005383  DEC      R3      ;DECREMENT COUNT, FINISHED?
1329      001968  001337  001337  BNE      0T48A      ;NO, TEST MORE DATA
1330
1331

```

```

1332
1333
1334
1335
1336
1337
1338 001212 104000
1339 001214 012703 000077
1340 001220 012704 000001
1341 001224 012737 174101 001036
1342
1343 001232 104000
1344 001234 013700 001022
1345 001240 012720 116000
1346 001244 005020
1347 001246 005020
1348 001250 013720 001036
1349 001254 012720 120000
1350 001260 005020
1351 001262 012710 172000
1352 001266 013777 001022 172556
1353 001274 004737 012340
1354
1355 001300 017700 172554
1356 001304 020400
1357 001306 001402
1358 001310 000000
1359 001322 000405
1360
1361 001324 005237 001036
1362 001326 005204
1363 001328 005303
1364 001324 001343

                                G491  SCOPE
                                MOV    #77,R3      ;SET UP EXECUTION COUNTER
                                MOV    #1,R4        ;SET UP COMPARED DATA
                                MOV    #175101,0SAVE ;SET UP BASIC FLOAD STATUS B"

                                G49A1  SCOPE
                                MOV    0BUF,R0     ;SET UP R0
                                MOV    #110000,(0)+ ;LOAD "POINT MODE"
                                CLR    (0)+         ;CLEAR X AXIS
                                CLR    (0)+         ;CLEAR Y AXIS
                                MOV    0SAVE,(0)+   ;LOAD "SET STATUS B"
                                MOV    #120000,(0)+ ;LOAD "SET GRAPHPLOT X MODE"
                                CLR    (0)+         ;LOAD "X GRAPHPLOT DATA"
                                MOV    #170000,(0)
                                MOV    0BUF,0OPC    ;LOAD THE DISPLAY P.C.
                                JSR     7,0LAY       ;EXECUTE A PROGRAM DELAY

                                MOV    #YR0,R0     ;READ Y AXIS
                                CMP    R4,R0        ;COMPARE TO EXPECTED VALUE
                                BEQ     ,+6          ;ARE THEY EQUAL?
                                HALT  ?            ;LOAD "STATUS B" FAILED TO LOAD
                                BR      0T00        ;THE Y AXIS CORRECTLY

                                INC     0SAVE       ;INCREMENT THE STATUS B COUNT
                                INC     R4          ;DECREMENT THE EXECUTION COUNT
                                DEC     R3          ;TEST MORE DATA
                                BNE     0T01A

```

```

1366
1367
1368
1369
1370
1371 001326 104000
1372 001330 012703 000077
1373 001334 012704 000001
1374 001336 012737 174101 001036
1375
1376 001346 104000
1377 001350 013700 001022
1378 001354 012720 116000
1379 001360 005020
1380 001362 005020
1381 001364 013720 001036
1382 001370 012720 124000
1383 001374 000000
1384 001376 012710 172000
1385 001402 013777 001022 172442
1386 001410 004737 012340
1387
1388 001424 017700 172436
1389 001430 042700 174000
1390 001424 020400
1391 001426 001402
1392 001430 000000
1393 001432 000405
1394
1395 001434 005237 001036
1396 001440 005204
1397 001442 005303
1398 001444 001341
1399
1400 001446 012777 174100 172346 G49B1 MOV    #175101,0BUF
1401 001454 012777 172000 172342 MOV    #170000,0BUF
1402 001462 013777 001022 172342 MOV    0BUF,0OPC

                                G49B1  SCOPE
                                MOV    #77,R3      ;SET UP EXECUTION COUNTER
                                MOV    #1,R4        ;SET UP COMPARED DATA
                                MOV    #175101,0SAVE ;SET UP BASIC FLOAD STATUS B"

                                G49BA1 SCOPE
                                MOV    0BUF,R0     ;SET UP R0
                                MOV    #110000,(0)+ ;LOAD "POINT MODE"
                                CLR    (0)+         ;CLEAR X AXIS
                                CLR    (0)+         ;CLEAR Y AXIS
                                MOV    0SAVE,(0)+   ;LOAD "SET STATUS B"
                                MOV    #120000,(0)+ ;LOAD "SET GRAPHPLOT Y MODE"
                                CLR    (0)+         ;LOAD "Y GRAPHPLOT DATA"
                                MOV    #170000,(0)
                                MOV    0BUF,0OPC    ;LOAD THE DISPLAY P.C.
                                JSR     7,0LAY       ;EXECUTE A PROGRAM DELAY

                                MOV    #XOPS,R0    ;READ X AXIS
                                BIC    0170000,R0  ;MASK TO BITS 0-9
                                CMP    R4,R0        ;COMPARE TO EXPECTED VALUE
                                BEQ     ,+6          ;ARE THEY EQUAL?
                                HALT  ?            ;LOAD "STATUS B" FAILED TO LOAD
                                BR      0T00        ;THE X AXIS CORRECTLY

                                INC     0SAVE       ;INCREMENT THE STATUS B COUNT
                                INC     R4          ;DECREMENT THE EXECUTION COUNT
                                DEC     R3          ;TEST MORE DATA
                                BNE     0T01A

```

14

14
14

```

1490      JEDGE FLAG TEST
1491      ITEST THAT EXCEEDING +Y AXIS SETS EDGE FLAG
1492
1493      027042 104000      GT541  SCOPE
1494      007044 013700 001022      MOV      08U1,R0
1495      027050 012720 116000      MOV      #110000,(0)+      ILOAD POINT
1496      007054 012720 000000      MOV      #0,(0)+      ILOAD X
1497      007060 013720 000000      MOV      08U1,R0      ILOAD Y
1498      007064 012720 110000      MOV      #110000,(0)+      ILOAD LONG VECTOR
1499      027070 012720 000000      MOV      #0,(0)+      ILOAD DELTA X
1500      027074 012720 000001      MOV      #1,(0)+      ILOAD DELTA Y
1501      007100 012720 172000      MOV      #170000,(0)+      ILOAD STOP
1502      027104 013777 001022 171740      MOV      08U1,00PC      ISTART DISPLAY
1503      027112 004737 012340      JSR      7,0LAY      IEXECUTE A PROGRAM DELAY
1504
1505      007116 032777 000040 171730      BIT      #40,00SR      ITEST BIT 5
1506      007124 001002      SNE      ,+4
1507      007126 000000      HALT      IEDGE FLAG FAILED TO SET
1508      007130 000424      BR      GT50
1509
1510      IEDGE FLAG TEST
1511      ITEST THAT THE EDGE FLAG CLEARS
1512
1513      027132 013700 001022      MOV      08U1,R0
1514      007136 012720 116000      MOV      #110000,(0)+      ILOAD POINT
1515      007140 012720 000000      MOV      #0,(0)+      ILOAD X
1516      007144 012720 000000      MOV      #0,(0)+      ILOAD Y
1517      007150 012720 172000      MOV      #170000,(0)+      ILOAD STOP
1518      027154 013777 001022 171666      MOV      08U1,00PC      ISTART DISPLAY
1519      027164 004737 012340      JSR      7,0LAY      IEXECUTE A PROGRAM DELAY
1520
1521      027170 032777 000040 171656      BIT      #40,00SR
1522      007176 001002      BCC      ,+4
1523      007200 000000      HALT      IERROR, EDGE FLAG FAILED TO CLEAR
1524
1525      IEDGE FLAG TEST
1526      ITEST THAT EXCEEDING -Y AXIS SETS EDGE FLAG
1527
1528      027202 104000      GT551  SCOPE
1529      007204 013700 001022      MOV      08U1,R0
1530      027210 012720 116000      MOV      #110000,(0)+      ILOAD POINT
1531      007214 012720 000000      MOV      #0,(0)+      ILOAD X
1532      007218 012720 000000      MOV      #0,(0)+      ILOAD Y
1533      007224 012720 110000      MOV      #110000,(0)+      ILOAD LONG VECTOR
1534      007230 012720 000000      MOV      #0,(0)+      ILOAD DELTA X
1535      007234 012720 000001      MOV      #1,(0)+      ILOAD DELTA Y
1536      007240 012720 172000      MOV      #170000,(0)+      ILOAD STOP
1537      027244 013777 001022 171600      MOV      08U1,00PC      ISTART DISPLAY
1538      027252 004737 012340      JSR      7,0LAY      IEXECUTE A PROGRAM DELAY
1539
1540      007256 032777 000040 171570      BIT      #40,00SR      ITEST BIT 5
1541      027264 001002      SNE      ,+4
1542      007266 000000      HALT      IEDGE FLAG FAILED TO SET
  
```

```

1544      ITEST THAT THE CHARACTER REGISTER IS LOADED PROPERLY
1545      I CODE 00
1546
1547      027270 104000      GT561  SCOPE
1548      007272 012777 100000 171022      MOV      #100000,08U1      ILOAD "CHARACTER MODE"
1549      007300 012777 000000 171516      MOV      #0,08U1F1      ILOAD "NULL" CHARACTER
1550      007306 012777 172000 171512      MOV      #170000,08U1F2
1551      027314 013777 001022 171530      MOV      08U1,00PC      ISTART DISPLAY
1552      027322 004737 012340      JSR      7,0LAY      IEXECUTE A PROGRAM DELAY
1553      007326 017700 171526      MOV      #170000,R0      IREAD CHARACTER REG.
1554      027332 042700 001777      BIC      #1777,R0      IMASK TO BITS 10-15
1555      027336 022700 000000      CMP      #0,R0
1556      027342 001401      BEQ      ,+4
1557      027344 000000      HALT      IERROR, CHARACTER REGISTER LOADED IN ERROR
1558
1559      ITEST THAT THE CHARACTER REGISTER IS LOADED PROPERLY
1560      I CODE 77
1561
1562      027346 104000      GT571  SCOPE
1563      027350 012777 100000 171444      MOV      #100000,08U1      ILOAD "CHARACTER MODE"
1564      027356 012777 000077 171440      MOV      #77,08U1F1      ILOAD CHARACTER
1565      007364 012777 172000 171434      MOV      #170000,08U1F2
1566      027372 013777 001022 171402      MOV      08U1,00PC      ISTART DISPLAY
1567      027400 004737 012340      JSR      7,0LAY      IEXECUTE A PROGRAM DELAY
1568      027404 017700 171450      MOV      #170000,R0      IREAD CHARACTER REG.
1569      027410 042700 001777      BIC      #1777,R0      IMASK TO BITS 10-15
1570      027414 022700 176000      CMP      #170000,R0
1571      027420 001401      BEQ      ,+4
1572      027422 000000      HALT      IERROR, CHARACTER REGISTER LOADED IN ERROR
1573
1574      ITEST THAT THE CHARACTER REGISTER IS LOADED PROPERLY
1575      I CODE 25
1576
1577      027424 104000      GT581  SCOPE
1578      027426 012777 100000 171366      MOV      #100000,08U1      ILOAD "CHARACTER MODE"
1579      007434 012777 000025 171362      MOV      #25,08U1F1      ILOAD CHARACTER
1580      007442 012777 172000 171356      MOV      #170000,08U1F2
1581      027450 013777 001022 171374      MOV      08U1,00PC      ISTART DISPLAY
1582      007456 004737 012340      JSR      7,0LAY      IEXECUTE A PROGRAM DELAY
1583      027462 017700 171372      MOV      #170000,R0      IREAD CHARACTER REG.
1584      027466 042700 001777      BIC      #1777,R0      IMASK TO BITS 10-15
1585      027472 022700 052000      CMP      #52000,R0
1586      027476 001401      BEQ      ,+4
1587      027500 000000      HALT      IERROR, CHARACTER REGISTER LOADED IN ERROR
  
```

```
1589
1590
1591
1592
1593 027502 104000
1594 027504 012777 100000 171310
1595 027512 012777 000052 171304
1596 027520 012777 172000 171300
1597 027526 013777 001022 171316
1598 027534 004737 012340
1599 027540 017700 171314
1600 027544 042700 001777
1601 027550 022700 124000
1602 027554 001401
1603 027556 000000
1604
1605
1606
1607
1608 007500 104000
1609 027562 012777 116000 171232
1610 027570 012777 001000 171286
1611 027576 012777 001000 171222
1612 027604 012777 100000 171216
1613 027612 005077 171214
1614 027616 012777 172000 171210
1615 027644 013777 001022 171200
1616 027632 004737 012340
1617
1618 027636 017700 171216
1619 027642 042700 001777
1620 027646 022700 000000
1621 027652 001402
1622 027654 000000
1623 027656 000417
1624
1625 007660 017700 171172
1626 027644 022700 001000
1627 027670 001402
1628 027672 000000
1629 027674 000410
1630
1631 027676 017700 171156
1632 027702 042700 170000
1633 027706 022700 001000
1634 027712 001401
1635 027714 000000
1636
```

TEST THAT THE CHARACTER REGISTER IS LOADED PROPERLY
CODE 52

GT591 SCOPE
MOV #100000,00BUF ;LOAD "CHARACTER MODE"
MOV #52,00BUF1 ;LOAD CHARACTER
MOV #170000,00BUF2
MOV 00UF7,00FC ;START DISPLAY
JSR 7,0LAY ;EXECUTE A PROGRAM DELAY
MOV #YPCB,R0 ;READ CHARACTER REG.
BIC #1777,R0 ;MASK TO BITS 10-10
CMP #120000,R0
BEQ ,+4
HALT ;ERROR, CHARACTER REGISTER LOADED IN ERROR

TEST THAT CHARACTER MODE DOES NOT HANG THE DISPLAY PROCESSOR
TEST THAT "NULL" DOES NOT CHANGE X OR Y AXIS

GT601 SCOPE
MOV #100000,00BUF ;POINT MODE
MOV #1000,00BUF1
MOV #1000,00BUF2
MOV #100000,00BUF3
CLR 00BUF4
MOV #170000,00BUF5
MOV 00UF7,00FC ;LOAD THE DISPLAY P.C.
JSR 7,0LAY ;EXECUTE A PROGRAM DELAY
MOV #YPCB,R0 ;READ CHARACTER REGISTER
BIC #1777,R0 ;MASK TO BITS 10-10
CMP #0,R0
BEQ ,+6
HALT ;CHARACTER REGISTER IN ERROR
BR GT61
MOV #YPCB,R0 ;READ X AXIS
CMP #1000,R0 ;ARE THEY EQUAL 7
BEQ ,+6
HALT ;"NULL" CHARACTER CHANGED X AXIS
BR GT61
MOV #YPCB,R0 ;READ Y AXIS
BIC #170000,R0 ;MASK TO BITS 0-9
CMP #1000,R0 ;ARE THEY EQUAL 7
BEQ ,+4
HALT ;"NULL" CHARACTER CHANGED Y AXIS

```
1636
1639
1640
1641
1642
1643 007716 104000
1644 027700 012777 116000 171074
1645 027726 012777 001000 171070
1646 027734 012777 001000 171044
1647 027742 012777 100000 171060
1648 027750 012777 000015 171004
1649 027756 012777 172000 171090
1650 027764 013777 001022 171060
1651 027772 004737 012340
1652
1653 027776 017700 171056
1654 010002 042700 001777
1655 010006 022700 032000
1656 010012 001402
1657 010014 000000
1658 010016 000417
1659
1660 010020 017700 171032
1661 010024 022700 000000
1662 010030 001402
1663 010032 000000
1664 010034 000410
1665
1666 010036 017700 171016
1667 010042 042700 170000
1668 010046 022700 001000
1669 010052 001401
1670 010054 000000
1671
```

TEST THAT CHARACTER MODE DOES NOT HANG THE DISPLAY PROCESSOR
TEST THAT "CR" DOES CHANGE X AND DOES NOT CHANGE Y AXIS

GT611 SCOPE
MOV #100000,00BUF ;POINT MODE
MOV #1000,00BUF1
MOV #1000,00BUF2
MOV #100000,00BUF3
MOV #1000,00BUF4
MOV #170000,00BUF5
MOV 00UF7,00FC ;LOAD THE DISPLAY P.C.
JSR 7,0LAY ;EXECUTE A PROGRAM DELAY
MOV #YPCB,R0 ;READ Y AXIS
BIC #1777,R0 ;MASK TO BITS 10-10
CMP #33000,R0
BEQ ,+6
HALT ;CHARACTER REGISTER FAILED TO LOAD CORRECTLY
BR GT62
MOV #YPCB,R0 ;READ X AXIS
CMP #0,R0 ;ARE THEY EQUAL 7
BEQ ,+6
HALT ;"CR" CHARACTER FAILED TO CHANGE X AXIS CORRECTLY
BR GT62
MOV #YPCB,R0 ;READ Y AXIS
BIC #170000,R0 ;MASK TO BITS 0-9
CMP #1000,R0 ;ARE THEY EQUAL 7
BEQ ,+4
HALT ;"CR" CHARACTER CHANGED Y AXIS

```

1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706

```

010056	104000	116000	170734	GT621	SCOPE		
010060	012777	001000	170730	MOV	#110000,00BUF	POINT MODE	
010066	012777	001000	170724	MOV	#1000,00BUF1		
010074	012777	001000	170720	MOV	#1000,00BUF2	1000,1000	
010102	012777	000012	170714	MOV	#100000,00BUF3	LOAD "CHARACTER MODE"	
010110	012777	001022	170710	MOV	#12,00BUF4		
010116	012777	001022	170700	MOV	#170000,00BUF5	LOAD THE DISPLAY P.C.	
010124	013777	012340		MOV	00UF,00PC	EXECUTE A PROGRAM DELAY	
010132	004737			JSR	7,0LAY		
010136	017700	001777		MOV	#YPC0,R0	READ CHARACTER REG.	
010142	022700	001402		BIC	#1777,R0	MASK TO BITS 10-15	
010146	000000			CHP	#2000,R0		
010152	000000			SEQ	,=0		
010154	000000			HALT		CHARACTER REGISTER IN ERROR	
010156	000477			BR	GT63		
010160	017700	001000		MOV	#XPC0,R0	READ X AXIS	
010164	022700	001000		CHP	#1000,R0	ARE THEY EQUAL ?	
010170	001402			SEQ	,=0	YES	
010172	000000			HALT		"LF" CHARACTER CHANGED X AXIS	
010174	000477			BR	GT63		
010176	017700	001777		MOV	#YPC0,R0	READ Y AXIS	
010202	022700	001000		BIC	#170000,R0	MASK TO BITS 10-15	
010206	001402			CHP	#1700,R0	ARE THEY EQUAL ?	
010212	001401			SEQ	,=0	YES	
010214	000000			HALT		"LF" CHARACTER FAILED TO CHANGED Y AXIS CORRECTLY	

```

1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741

```

010216	104000	116000	170574	GT62A1	SCOPE		
010220	012777	000000	170570	MOV	#110000,00BUF	POINT MODE	
010226	012777	000000	170564	MOV	#0,00BUF1		
010234	012777	000000	170560	MOV	#1000,00BUF2	10,1000	
010242	012777	000012	170554	MOV	#100000,00BUF3	LOAD "CHARACTER MODE"	
010250	012777	001022	170550	MOV	#10,00BUF4	LOAD AN "A"	
010256	012777	001022	170540	MOV	#170000,00BUF5	LOAD THE DISPLAY P.C.	
010264	013777	001022	170530	MOV	00UF,00PC	EXECUTE A PROGRAM DELAY	
010272	004737	012340		JSR	7,0LAY		
010276	017700	001777		MOV	#YPC0,R0	READ CHARACTER REG	
010302	042700	001777		BIC	#1777,R0	MASK TO BITS 10-15	
010306	022700	002000		CHP	#2000,R0		
010312	001402			SEQ	,=0		
010314	000000			HALT		CHARACTER REGISTER IN ERROR	
010316	000417			BR	GT63		
010320	017700	001000		MOV	#XPC0,R0	READ X AXIS	
010324	023700	001000		CHP	#000000,R0	ARE THEY EQUAL ?	
010330	001402			SEQ	,=0	YES	
010332	000000			HALT		"A" CHARACTER FAILED TO CHANGED X AXIS CORRECTLY	
010334	000410			BR	GT63		
010336	017700	001777		MOV	#YPC0,R0	READ Y AXIS	
010342	042700	001000		BIC	#170000,R0	MASK TO BITS 10-15	
010346	022700	001000		CHP	#1000,R0	ARE THEY EQUAL ?	
010352	001401			SEQ	,=0	YES	
010354	000000			HALT		"A" CHARACTER CHANGED Y AXIS	

```

1743
1744
1745
1746
1747
1748 012356 104000 116000 170434 G7631 SCOPE
1749 012360 012777 001000 170430 MOV 011000,000UF ;POINT MODE
1750 012366 012777 001000 170430 MOV 011000,000UF1 ;
1751 012374 012777 001000 170424 MOV 011000,000UF2 ;1000,1000
1752 012402 012777 100000 170420 MOV 011000,000UF3 ;LOAD "CHARACTER MODE"
1753 012410 012777 000010 170414 MOV 0110,000UF4 ;
1754 012416 012777 172000 170410 MOV 011000,000UF5 ;
1755 012424 013777 001022 170420 MOV 000UF0PC ;LOAD THE DISPLAY P.C.
1756 012432 004737 012340 JSR 7,0LAY ;EXECUTE A PROGRAM DELAY
1757
1758 012436 017700 170416 MOV 0YPOS,R0 ;READ CHARACTER REG
1759 012442 042700 001777 BIC 01777,R0 ;MASK TO BITS 10-15
1760 012446 022700 020000 CMP 020000,R0 ;
1761 012452 001402 BEQ 104 ;
1762 012454 000000 HALT ;CHARACTER REGISTER IN ERROR
1763 012456 000425 BR G763
1764
1765 012460 017700 170372 MOV 0YPOS,R0 ;READ X AXIS
1766 012464 023700 001050 CMP 001050,R0 ;ARE THEY EQUAL ?
1767 012470 001402 BEQ 104 ;YES
1768 012472 000000 HALT ;"BB" CHARACTER FAILED TO CHANGE X AXIS CORRECTLY
1769 012474 000416 BR G764
1770
1771 012476 017700 170356 MOV 0YPOS,R0 ;READ Y AXIS
1772 012502 042700 170000 BIC 017000,R0 ;MASK TO BITS 0-9
1773 012506 022700 001000 CMP 01000,R0 ;ARE THEY EQUAL ?
1774 012512 001401 BEQ 104 ;YES
1775 012514 000000 HALT ;"BB" CHARACTER CHANGED Y AXIS
1776
1777
1778
1779 012516 017700 170332 G763A1 MOV 000R,R0 ;READ STATUS
1780 012522 032700 000100 BIT 0100,R0 ;
1781 012526 001401 BEQ 104 ;
1782 012530 000000 HALT ;SHIFT OUT STATUS BIT IS SET
1783

```

```

1785
1786
1787
1788
1789 012532 104000 116000 170260 G7641 SCOPE
1790 012534 012777 001000 170254 MOV 011000,000UF ;POINT MODE
1791 012542 012777 001000 170250 MOV 011000,000UF1 ;
1792 012550 012777 001000 170250 MOV 011000,000UF2 ;1000,1000
1793 012556 012777 100000 170244 MOV 011000,000UF3 ;LOAD "CHARACTER MODE"
1794 012564 012777 037416 170240 MOV 037416,000UF4 ;"SHIFT-OUT" IN LOW BYTE 077 IN HIGH BYTE
1795 012572 012777 172000 170234 MOV 011000,000UF5 ;LOAD STOP
1796 012600 013777 001022 170244 MOV 000UF0PC ;START DISPLAY
1797 012606 004737 012340 JSR 7,0LAY ;EXECUTE A PROGRAM DELAY
1798
1799 012612 017700 170242 MOV 0YPOS,R0 ;READ CHARACTER REG
1800 012616 042700 001777 BIC 01777,R0 ;MASK TO BITS 10-15
1801 012622 022700 170000 CMP 017000,R0 ;
1802 012626 001402 BEQ 104 ;
1803 012630 000000 HALT ;CHARACTER REGISTER IN ERROR
1804 012632 000426 BR G762 ; AFTER A SHIFT-OUT COMMAND
1805
1806 012634 017700 170214 MOV 000R,R0 ;READ STATUS REGISTER
1807 012640 032700 000100 BIT 0100,R0 ;
1808 012644 001002 BNE 104 ;
1809 012646 000000 HALT ;SHIFT OUT STATUS BIT FAILED TO SET
1810 012650 000417 BR G763
1811
1812 012652 017700 170200 MOV 0YPOS,R0 ;READ X POS
1813 012656 022700 001000 CMP 01000,R0 ;
1814 012662 001402 BEQ 104 ;
1815 012664 000000 HALT ;SHIFT-OUT CHARACTER CHANGED X AXIS
1816 012666 000410 BR G762
1817
1818 012670 017700 170164 MOV 0YPOS,R0 ;READ Y POS
1819 012674 042700 170000 BIC 017000,R0 ;MASK
1820 012700 022700 001000 CMP 01000,R0 ;
1821 012704 001401 BEQ 104 ;
1822 012706 000000 HALT ;SHIFT-OUT CHARACTER CHANGED Y AXIS

```

```

1824
1825
1826
1827
1828 010710 104000
1829 010712 000005
1830 010714 005003
1831 010716 012777 100000 170076
1832 010724 012777 000016 170072
1833 010732 012777 172000 170066
1834 010740 110337 013467
1835 010744 000240
1836 010746 013777 001022 170076
1837 010754 004737 012340
1838
1839 010760 032777 000100 170066
1840 010766 001402
1841 010770 000000
1842 010772 000407
1843
1844 010774 005203
1845 010776 022703 000017
1846 011002 001774
1847 011004 022703 000040
1848 011010 001333
1849
1850
1851
1852
1853
1854 011012 104000
1855 011014 000005
1856 011016 012777 100000 167774
1857 011024 012777 000016 167772
1858 011032 012777 172000 167766
1859 011040 112737 000040 013467
1860 011044 000240
1861 011050 013777 001022 167774
1862 011056 004737 012340
1863
1864 011002 032777 000100 167764
1865 011070 001002
1866 011072 000000
1867 011074 000407
1868

```

TEST THAT "SHIFT-OUT" DOES NOT GENERATE A STATUS BIT
IF "SHIFT-OUT" FOLLOWED BY CODE 0 THRU 37 EXCEPT #17)

GT651: SCOPE
RESET
CLR R3
MOV #100000,00BUF
MOV #16,00BUF1
MOV #170000,00BUF2
GT65A1: MOV# R3,BUFFER#3
NOP
MOV 00BUF#00RC
JSR 7,0LAY
BIT #100,00SR
BEO #4
HALT
BR GT60
GT65B: INC R3
CMP #17,R3
BEO GT60B
CMP #40,R3
BNE GT60A
TEST FOR "SHIFT-IN"
TEST FOR #40
IS IT #40
YES, NEXT TEST

TEST THAT "SHIFT-OUT" FOLLOWED BY CODE 40 GENERATE A
SHIFT STATUS BIT

GT661: SCOPE
RESET
MOV #100000,00BUF
MOV #16,00BUF1
MOV #170000,00BUF2
GT66A1: MOV# #40,BUFFER#3
NOP
MOV 00BUF#00RC
JSR 7,0LAY
BIT #100,00SR
BNE
HALT
BR GT67
TEST "SHIFT-OUT" STATUS BIT
ION CHARACTER IN R3

GT661: SCOPE
RESET
MOV #100000,00BUF
MOV #16,00BUF1
MOV #170000,00BUF2
GT66A1: MOV# #40,BUFFER#3
NOP
MOV 00BUF#00RC
JSR 7,0LAY
BIT #100,00SR
BNE
HALT
BR GT67
TEST "SHIFT-OUT" STATUS BIT
ION CHARACTER IN R3

```

1870
1871 011076 000005
1872 011100 032777 000100 167746
1873 011106 001402
1874 011110 000000
1875 011122 000400
1876
1877
1878
1879
1880 011114 104000
1881 011116 012777 100000 167676
1882 011124 005077 167674
1883 011130 012777 007000 167606
1884 011136 012777 000040 167662
1885
1886 011144 012777 172000 167656
1887 011152 000005
1888 011154 000240
1889 011156 013777 001022 167666
1890 011144 004737 012340
1891
1892 011170 032777 000100 167656
1893 011176 001002
1894 011200 000000
1895 011202 000410
1896
1897 011204 017700 167650
1898 011210 042700 001777
1899 011214 022700 100000
1900 011220 001401
1901 011222 000000
1902
1903
1904

```

RESET
BIT
BEO
HALT
BR
GT67
TEST SHIFT-OUT BIT
IF CLEARED
SHIFT OUT STATUS BIT FAILED TO CLEAR

TEST THAT "SHIFT-OUT" IN THE HIGH BYTE FOLLOWED BY A CHARACTER
IN THE NEXT LOW BYTE GENERATES A STATUS BIT

GT671: SCOPE
MOV
CLR
MOV
MOV
MOV
RESET
NOP
MOV
MOV
JSR
BIT
BNE
HALT
BR
MOV
BIC
CMP
BEO
HALT
#100000,00BUF
#00BUF1
#7000,00BUF1
#40,00BUF2
#170000,00BUF3
00BUF#00RC
7,0LAY
#100,00SR
#4
GT60
#0003,R0
#1777,R0
#100000,R0
#4
LOAD SET "CHAR" MODE
LOAD "SHIFT-OUT" INTO THE LOW BYTE
LOAD "SHIFT-OUT" INTO THE HIGH BYTE
LOAD A "SHIFT-OUT" CHARACTER IN THE NEXT
INIOR <LOW BYTE>
START THE DISLAY
EXECUTE A PROGRAM DELAY
TEST THE STATUS REGISTER
SHIFT-OUT IN THE HIGH BYTE FAILED TO
SET A STATUS BIT
READ Y POS
HACK TO BITS 10-18
TEST FOR CHAR #40
CHARACTER REGISTER IN ERROR AFTER A
"SHIFT-OUT" <HIGH BYTE> FOLLOWED BY
#40 <LOW BYTE NEXT WORD>

```

1926                                ;STOP INTERRUPT TEST
1927                                ;TEST FOR NO INTERRUPT
1928
1929                                G768:  SCOPE
1930                                RESET
1931                                MOV     #0788A,9000NE  ;LOAD RETURN FROM DONE INTERRUPT
1932                                MOV     #0788A,9TIMEVT  ;LOAD RETURN FROM TIME-OUT INTERRUPT
1933                                MOV     #0788A,9LPVCT  ;LOAD RETURN FROM LIGHT-PEN INTERRUPT
1934                                MOV     #160000,00SUP  ;LOAD "NOISBLAY NOP"
1935                                MOV     #173000,00SUP1 ;LOAD "STATUS A"-"STOP"-"STOP INT", ENABLE"
1936                                CLR     0000H          ;LOWER MACHINE PRIORITY
1937                                MOV     00UE,00PC      ;LOAD OIBLAY P.O.
1938                                NOP
1939                                NOP
1940                                NOP
1941                                NOP
1942                                BR      ,+4
1943
1944                                G768:  HALT                ;GT-48 INTERRUPTED IN ERROR
1945
1946                                ;STOP INTERRUPT TEST
1947                                ;TEST FOR INTERRUPT
1948
1949                                G769:  SCOPE
1950                                RESET
1951                                MOV     #0788A,9000NE  ;LOAD RETURN ADDRESS FROM INTERRUPT
1952                                MOV     #0788B,9LPVCT  ;LOAD LP VECTOR
1953                                MOV     #0788C,9TIMEVT  ;LOAD TO VECTOR
1954                                MOV     #160000,00SUP  ;LOAD "NOISBLAY NOP"
1955                                MOV     #173000,00SUP1 ;LOAD "STATUS A"-"STOP"-"STOP INT", ENABLE-INT"
1956                                CLR     0000H          ;LOWER MACHINE PRIORITY
1957                                MOV     00UE,00PC      ;LOAD OIBLAY P.O.
1958                                NOP
1959                                NOP
1960                                NOP
1961                                NOP
1962                                HALT
1963                                G769A:  MOV     0000NE,9000NE  ;GT-48 FAILED TO GENERATE 4 STOP INTERRUPT
1964                                CMP     (SP)+,(SP)+
1965                                BR      0778
1966
1967                                G769B:  CMP     (SP)+,(SP)+
1968                                HALT
1969
1970                                BB      0778
1971
1972                                G769C:  CMP     (SP)+,(SP)+
1973                                HALT
1974
1975                                ;GT-48 STOP (00NE) INTERRUPTED TO THE
1976                                ;LIGHT-PEN VECTOR
1977
1978                                ;GT-48 STOP (00NE) INTERRUPTED TO THE
1979                                ;TIME-OUT VECTOR
1980

```

```

1956                                ;SHIFT OUT INTERRUPT TEST
1957                                ;TEST FOR INTERRUPT
1958
1959                                G770:  SCOPE
1960                                RESET
1961                                MOV     #0788B,9000NE  ;LOAD 00NE VECTOR
1962                                MOV     #0788C,9LPVCT  ;LOAD LIGHT-PEN VECTOR
1963                                MOV     #0788A,9TIMEVT  ;LOAD RETURN ADDRESS
1964                                MOV     #180000,00SUP  ;LOAD "CHARACTER MODE"
1965                                MOV     #200016,900UF1  ;LOAD "SHIFT-OUT"
1966                                MOV     #173000,00SUP2 ;LOAD "SHIFT-OUT"
1967                                CLR     0000H          ;START OIBLAY
1968                                MOV     00UE,00PC
1969                                NOP
1970                                NOP
1971                                NOP
1972                                NOP
1973                                NOP
1974                                HALT
1975                                G770A:  MOV     9TIMEVT,9TIMEVT
1976                                CMP     (SP)+,(SP)+
1977                                BR      0778
1978
1979                                G770B:  CMP     (SP)+,(SP)+
1980                                HALT
1981
1982                                BB      0778
1983
1984                                G770C:  CMP     (SP)+,(SP)+
1985                                HALT
1986
1987                                ;GT-48 SHIFT-OUT INTERRUPTED TO THE
1988                                ;LIGHT PEN VECTOR
1989

```

```
1990
1991
1992
1993
1994 011500 104000
1995 011532 000000
1996 011534 013777 001064 167303
1997 011562 013777 001070 167276
1998 011570 012777 011616 167274
1999 011576 000077 167216
2000 011602 012777 177776 167242
2001 011630 004737 012340
2002 011614 000000
2003
2004 011616 000240
2005 011630 013777 001074 167244
2006 011626 022626
2007
2008
2009
2010 011630 104000
2011 011632 000000
2012 011634 012777 011676 167224
2013 011642 012777 130140 167132
2014 011630 012777 173000 167146
2015 011636 000077 167136
2016 011662 013777 001022 1A7162
2017 011670 004737 012340
2018 011674 000401
2019 011676 000000
2020 011700 013777 001070 1A7160

TIME-OUT INTERRUPT TEST
GT71: SCOPE
RESET
MOV 000NE,000NE
MOV LPVCT1,9LPVCT
MOV #GT71A,9TIMEVT
CLR 0PSN
MOV #177776,0OPC
JSR 7,0LAT
HALT
;LOAD RETURN ADDRESS
;LOAD DISPLAY P.O.
;EXECUTE A PROGRAM DELAY
;GT-40 FAILED TO INTERRUPT ON TIME-OUT

GT71A: NOP
MOV TIMEVT,9TIMEVT
CMP (SP)+,(SP)+

INO LIGHT PEN INTERRUPT TEST
GT72: SCOPE
RESET
MOV #GT72A,9LPVCT
MOV #100140,00BUP
MOV #173000,00BUP1
CLR 0PSN
MOV 00BUP,0OPC
JSR 7,0LAT
BR 104
GT72A: HALT
MOV LPVCT1,9LPVCT
;LOAD RETURN ADDRESS
;LOAD DISPLAY BUFFER
;EXECUTE A PROGRAM DELAY
;GT-40 INTERRUPTED ON FALSE LIGHT PEN FLAG
```

```
2022
2023
2024 011706 042737 177437 001004
2025 011714 001001
2026 011716 000000
2027 011720 022737 000340 001304
2028 011726 001001
2029 011730 000000
2030
2031 011732 013737 001004 011706
2032 011740 162737 000040 011756
2033 011746 013737 001004 011760
2034 011754 000402
2035
2036 011756 000140
2037 011760 000200
2038
2039
2040
2041
2042 011762 104000
2043 011764 000000
2044 011766 042777 011030 167046
2045 011774 012777 173400 167020
2046 011802 013777 011756 167010
2047 011810 013777 001022 167034
2048 011816 000240
2049 011820 000240
2050 011822 000240
2051 011824 000240
2052 011826 000000
2053
2054 011830 022626
2055
2056
2057
2058
2059 011832 104000
2060 011834 000000
2061 011836 012777 012100 167016
2062 011844 012777 173400 16A750
2063 011852 013777 011760 166748
2064 011860 013777 001022 166764
2065 011866 000240
2066 011870 000240
2067 011872 000240
2068 011874 000240
2069 011876 000401
2070 011880 000000
2071 011882 013777 001064 166752
2072 011884 000000
2073 011886 000000

PRE BR LEVEL SETUP
BIC #177437,0SPBR
BNE 104
HALT
CMP #340,00PSN
SNE 104
HALT
;MARK TO BITS
;BR LEVEL WAS 8
;BR LEVEL WAS 9

BRLEV1: 140
BRLEV2: 200

BR LEVEL TEST (BR=1)
;TEST FOR INTERRUPT
GT73: SCOPE
RESET
MOV #GT73A,0000NE
MOV #173400,00BUP
MOV 9RLEV1,0PSN
MOV 00BUP,0OPC
NOP
NOP
NOP
NOP
HALT
;LOAD RETURN ADDRESS
;LOAD "STATUS A"-NO INTERRUPT ENABLE
;LOAD THE DISPLAY P.O.
;NO STOP INTERRUPT ON BR LEVEL INDICATED =1
;CHECK TO SEE IF PROPER BR LEVEL

GT73A: CMP (SP)+,(SP)+

BR LEVEL TEST (BR)
;TEST THAT THE GT-40 DOES NOT INTERRUPT AT THE LEVEL INDICATED
GT74: SCOPE
RESET
MOV #GT74A,0000NE
MOV #173400,00BUP
MOV 9RLEV2,0PSN
MOV 00BUP,0OPC
NOP
NOP
NOP
NOP
BR 104
GT74A: HALT
MOV 000NE,000NE
RESET
;LOAD RETURN ADDRESS
;LOAD "STATUS A"-STOP-STOP INT ENABLE
;LOWER MACHINE PRIORITY TO INDICATED LEVEL
;NEXT TEST
;GT-40 INTERRUPTED ON THE WRONG BR LEVEL
;LOAD INTERRUPT VECTOR
```



```

2169
2170 012464 012777 000340 166326 START1: MOV #340,0PSW
2171 012472 012786 000500 MOV #STRPTR,SP
2172 012476 004737 001076 JSR PC,SETUP
2173 012502 012701 012464 MOV #START1,R1
2174 012506 012777 012542 166346 MOV #SPACE,0000NE ISET UP DONE VECTOR
2175 012514 013777 001004 166342 MOV #SPR,0000NE1
2176 012522 012777 012616 166336 MOV #SPACE,0LPVCT ISET UP LIGHT=PEN VECTOR
2177 012530 013777 001004 166332 MOV #SPR,0LPVCT1
2178 012536 000240 NOP
2179 012540 000240 NOP
2180 012542 012786 000500 SPCEA: MOV #STRPTR,3P
2181 012546 012737 173400 013426 MOV #0010P,FRM7A
2182 012554 017700 166260 MOV #SWR,R0
2183 012560 000100 ROL R0
2184 012562 042700 177761 BIC 0:77761,R0
2185 012566 010002 012636 MOV #SP[CH(0),R2
2186 012572 010277 166254 MOV R2,00PC ISTART THE DISPLAY
2187
2188 012576 000240 NOP
2189 012600 000240 NOP
2190 012602 005077 166212 SPCEC: CLR 0PSW
2191 012606 000001 WAIT
2192 012610 000240 NOP
2193 012612 000240 NOP
2194 012614 000752 BR SPCEA
2195
2196 012616 012737 164000 013426 SPACEB: MOV #000P,FRM7A
2197 012624 012777 000001 166220 MOV #1,00PC I SINGLE STEP THE DISPLAY
2198 012632 022626 CMP #SP1,1SP)
2199 012634 000762 BR SPCEC
2200
2201
2202 012636 012656 OSPTCH: FRAME0
2203 012640 012704 FRAME1
2204 012642 012732 FRAME2
2205 012644 012760 FRAME3
2206 012646 013000 FRAME4
2207 012650 013044 FRAME5
2208 012652 013132 FRAME6
2209 012654 013356 FRAME7
2210
2211

```

```

2213
2214
2215 012606 117004 FRAME0: POINT:INT4,LINE0
2216 012610 041000 INTX+1000
2217 012612 000600 000
2218 012614 041000 INTX+1000
2219 012616 000600 000
2220 012618 112400 LONGV,INT2
2221 012622 040600 INTX+000
2222 012624 000000 0
2223 012626 173400 OSTOP
2224 012628 160000 OJMP
2225 012630 012656 FRAME0
2226
2227 012704 117004 FRAME1: POINT:INT4,LINE0
2228 012706 040200 INTX+200
2229 012710 000600 000
2230 012712 041000 INTX+1000
2231 012714 000600 000
2232 012716 112400 LONGV,INT2
2233 012720 040600 INTX+MINUSX+000
2234 012722 000000 0
2235 012724 173400 OSTOP
2236 012726 160000 OJMP
2237 012730 012704 FRAME1
2238
2239 012732 117004 FRAME2: POINT:INT4,LINE0
2240 012734 041000 INTX+1000
2241 012736 001200 1200
2242 012740 041000 INTX+1000
2243 012742 000600 000
2244 012744 112400 LONGV,INT2
2245 012746 040000 INTX
2246 012750 000400 000
2247 012752 173400 OSTOP
2248 012754 160000 OJMP
2249 012756 012732 FRAME2
2250
2251 012740 117004 FRAME3: POINT:INT4,LINE0
2252 012762 041000 INTX+1000
2253 012764 000200 200
2254 012766 041000 INTX+1000
2255 012770 000600 000
2256 012772 112400 LONGV,INT2
2257 012774 040000 INTX
2258 012776 040400 MINUSX+400
2259 012800 173400 OSTOP
2260 012802 160000 OJMP
2261 012804 012760 FRAME3
2262

```

```

2264
2265 013006 117004
2266 013010 000000
2267 013012 000000
2268 013014 110000
2269 013016 041777
2270 013020 000000
2271 013022 040000
2272 013024 001377
2273 013026 061777
2274 013030 000000
2275 013032 040000
2276 013034 021377
2277 013036 173400
2278 013040 160000
2279 013042 013006
2280
2281 013044 117004
2282 013046 000740
2283 013050 000540
2284 013052 104000
2285 013054 057600
2286 013056 057677
2287 013060 040077
2288 013062 077677
2289 013064 077600
2290 013066 077777
2291 013070 040177
2292 013072 057777
2293 013074 114000
2294 013076 057600
2295 013080 000600
2296 013082 130000
2297 013084 047600
2298 013086 047637
2299 013090 040037
2300 013092 067437
2301 013094 067600
2302 013096 067737
2303 013098 040137
2304 013102 047737
2305 013104 173400
2306 013106 160000
2307 013130 013044
2308

```

```

FRAME4 POINT:INT4:LINE0
0
0
LONGV
INTX:MAXX
0
INTX
MAXV
INTX:MINUSX:MAXX
0
INTX
MINUSX:MAXV
DSTOP
DJMP
FRAME4

FRAME5 POINT:INT4:LINE0
740
540
SHORTV
INTX:17600
INTX:17677
INTX:77
INTX:MINUSX:17477
INTX:MINUSX:17600
INTX:MINUSX:17777
INTX:177
INTX:17777
POINT
760
020
RELATV
INTX:7400
INTX:7437
INTX:37
INTX:MINUSX:7637
INTX:MINUSX:7600
INTX:MINUSX:7737
INTX:137
INTX:7737
DSTOP
DJMP
FRAME5

```

```

2310
2311 013132 117004
2312 013134 000200
2313 013136 000600
2314 013140 100000
2315
(1) 013142 040500 041502 040504
(1) 013150 043506 044510 049512
(1) 013156 046514 047516 050520
(1) 013164 051522 052524 053526
(1) 013172 054530 132
(1) 013175 040 021041 022043
(1) 013202 023045 024047 025051
(1) 013210 026053 027055 030057
(1) 013216 031061 032063 033065
(1) 013224 034067 035071 036053
(1) 013232 037075 000077
(1)
2316 013236 114000
2317 013240 000200
2318 013242 000540
2319 013244 100000
2320
(1) 013246 140 141 142
(1) 013251 143 144 145
(1) 013254 146 147
(1) 013256 150 151 152
(1) 013261 153 154 155
(1) 013264 156 157
(1) 013266 160 161 142
(1) 013271 163 164 165
(1) 013274 166 167
(1) 013276 170 171 172
(1) 013301 173 174 175
(1) 013304 176 177
2321
(1) 013306 016 000 001
(1) 013311 002 003 004
(1) 013314 005 006 007
(1) 013317 010 011 012
(1) 013322 013 014 015
(1) 013325 016
(1) 013326 020 021 022
(1) 013331 023 024 025
(1) 013334 026 027 028
(1) 013337 031 032 033
(1) 013342 034 035 036
(1) 013345 037 017 000
(1)
2322 013350 173400
2323 013352 160000
2324 013394 013132
2325

```

```

FRAME6 POINT:INT4:LINE0
200
600
CHAR
,ASCII '0ABCDEFGHIJKLMNPQRSTUVWXYZ!
,ASCII " '!"#$%&'()*+,-./0123456789:;<=>?@
,EVEN
POINT
200
540
CHAR
, BYTE 140,141,142,143,144,145,146,147
, BYTE 150,151,152,153,154,155,156,157
, BYTE 160,161,162,163,164,165,166,167
, BYTE 170,171,172,173,174,175,176,177
, BYTE 16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,17,0
,EVEN
DSTOP
DJMP
FRAME6

```

```

2327
2328 013356 117004 FRAME7) POINTINT4LINE0
2329 013360 000400 400
2330 013362 000700 700
2331 013364 170200 STAYSAILPLIJE
2332 013366 110140 LONGVILPON
2333 013370 041000 INTX+1000
2334 013372 000000 0
2335 013374 114000 POINT
2336 013376 000400 400
2337 013400 000600 600
2338 013402 170300 STAYSAILPDARK
2339 013404 110140 LONGVILPON
2340 013406 041000 INTX+1000
2341 013410 000000 0
2342 013412 114000 POINT
2343 013414 000400 400
2344 013416 000500 500
2345 013420 110100 LONGVILPOFF
2346 013422 041000 INTX+1000
2347 013424 000000 0
2348 013426 173400 PRH7A) DSTOP
2349 013430 114000 POINT
2350 013432 000700 700
2351 013434 001000 1000
2352 013436 100000 CHAR
2353 013440 044107 024524 .ASCIZ /LIGHT=SEN HIT/
      013446 042520 020116 044510
      013454 000124
2354
2355 013456 173400 .EVEN
2356 013460 160000 DSTOP
2357 013462 013356 GJMR
2358
2359 013464 000000 FRAME7
2360
2361 000001 BUFFER) 0
      .ENO

```

RLKOFF = 000020	BLKDN = 000030	BRLV1 011750	BRLV2 011760
BUFFER 013466	CHAR = 100000	OH01ZE 001050	ENTR 001044
DBUF 001050	DBUF1 001054	DBUF2 001056	DBUF3 001058
DBUF4 001052	DBUF5 001054	ODONE 001058	DBONE1 001060
D190LA = 177570	DJMP = 100000	DLAY 013340	OLATA 012344
DLAY1 012352	DLAY1A 012350	ONDP = 100000	ODCORE 001250
DPC 001052	DSAVE 001056	OSFOR 001054	OSPTCH 012630
DNR 001054	DSTOP = 173400	END 001204	FRAME5 012696
FRAME1 012700	FRAME2 012732	FRAME3 012760	FRAME6 013000
FRAME5 013044	FRAME6 013130	FRAME7 013386	PRH7A 013426
GRAPHX = 120000	GRAPHY = 104000	GREEN = 000002	GEADD 001000
GSCMSZ 001006	GLFSZ 000010	GOSENO 000014	GSVCT 001002
GSYAXS 001012	GTBUSS 001014	GTFC 001012	GTPCA 001456
GT 011532	GT1 002422	GT10 002446	GT11 002310
GT12 002356	GT13 002432	GT14 002460	GT15 002020
GT16 002364	GT17 002432	GT18 002470	GT19 002740
GT2 001624	GT20 003006	GT21 003030	GT22 003116
GT23 003340	GT24 003226	GT25 003272	GT26 003330
GT27 003420	GT28 003474	GT29 003540	GT3 001664
GT30 003620	GT31 003672	GT32 003776	GT33 004060
GT34 004356	GT35 004354	GT36 004392	GT37 004490
GT38 004556	GT39 004604	GT39A 004670	GT4 001726
GT40 004770	GT40A 005010	GT41 005112	GT42 005210
GT43 005306	GT43A 005326	GT44 005430	GT44A 005490
GT45 005552	GT46 005600	GT47 005740	GT47A 005766
GT48 005796	GT48A 006110	GT49 006212	GT49A 006234
GT5 001770	GT50 006326	GT50A 006350	GT50B 006444
GT51 006470	GT52 006562	GT53 006702	GT54 006740
GT55 006720	GT56 006770	GT57 007346	GT58 007426
GT59 007502	GT6 002026	GT60 007560	GT61 007710
GT62 010056	GT62A 010216	GT63 010356	GT63A 010916
GT64 010030	GT65 010710	GT69A 010740	GT68 010774
GT66 011010	GT66A 011040	GT67 011114	GT69 011224
GT68A 011312	GT69 011314	GT69A 011402	GT70 011414
GT69C 011022	GT7 002072	GT70 011426	GT70A 011522
GT70R 011534	GT70C 011544	GT71 011600	GT71A 011616
GT72 011630	GT72A 011670	GT73 011702	GT73A 012030
GT74 012032	GT74A 012100	GT78 012114	GT8 002130
GT9 012202	HERE 012226	HERE1 012250	WLOPHR 012414
ICNT 001016	INCR = 000100	INTX = 040000	INT5 = 002300
INT1 = 002200	INT2 = 002400	INT3 = 002400	INT4 = 002000
INT5 = 003200	INT6 = 003400	INT7 = 003400	ITALD = 000040
ITAL1 = 000060	LFSIZE 001046	LINE5 = 000004	LINE1 = 000000
LINE2 = 000006	LINE3 = 000007	LOICA 002440	LONGV = 110000
LOWPWR 012364	LOWSV 012440	LPOARK = 000300	LPITE = 000200
LPVCT = 000100	LPON = 000140	LPVCT 001066	LPVOT1 000070
MAXSX = 001760	MAXSY = 000077	MAXX = 001777	MAXY = 001377
MINBUY = 000500	MINUSX = 000000	MINUSY = 000000	MC = 000000
PCINT = 114000	PW 001020	RD = 000003	RELATV = 130000
R0 = 000000	R1 = 000001	R2 = 000002	R3 = 000003
R4 = 000004	R5 = 000005	SCOPE = 100000	SCOREA 012320
SCORFB 012332	SETUP 001076	SETUPA 001106	SETURB 001130
SHORTV = 114000	SIFE 001042	SIZE0 = 000400	SIBEL = 000000
SP = 000006	SPACEB 012016	SPECA 012342	SPREC 012600

GT=42/RT=44 INSTRUCTION TEST II MAINDEC=11-DDGTB=B MACY11,024 25=NOV=73 10150 PAGE 93.2
DDGTB=011 SYMBOL TABLE

START	001336	START1	012464	STATSA	= 170000	STATSB	= 174000
SYKPTR	= 000000	SWR	001040	SYNOFF	= 000010	SYNBN	= 000014
TIMEVT	001070	TIMEVT1	001074	TPCSR	= 177004	TPDBR	= 177006
VPOS	001000	VPOS	001060	.	= 011460		

ERRORS DETECTED: 0

REM *

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DDGTC-A-D
PRODUCT NAME:	GT48/GT44 VISUAL DISPLAY TEST WITH VR17 DISPLAY
DATE CREATED:	NOVEMBER 11 1973
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	RAYMOND SHOOP

COPYRIGHT (C) 1973, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED TO PURCHASER UNDER A LICENSE FOR USE
ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION
OF DEC'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM; EXCEPT
AS MAY OTHERWISE BE PROVIDED IN WRITING BY DEC.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT
NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL
EQUIPMENT CORPORATION.

DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF
ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.

1. ABSTRACT

THIS PROGRAM CONTAINS A SERIES OF PATTERNS THAT ARE USED AS AIDS IN THE ALIGNMENT AND ADJUSTMENT OF THE GT40/GT44 DISPLAY SYSTEM WITH A VR17. FOR THIS TEST THE MAINTENANCE SWITCHES ARE NOT USED (NORMAL POSITION).

2. REQUIREMENTS

2.1 EQUIPMENT

GT40 DISPLAY SYSTEM WITH VR17 DISPLAY SCOPE OR
GT44 DISPLAY SYSTEM WITH VR17 DISPLAY SCOPE.

2.2 STORAGE

THIS PROGRAM USES LESS THAN 4K OF MEMORY.

2.3 PRELIMINARY PROGRAMS

ALL PROCESSOR MAINDECS, GT40/GT44 INSTRUCTION TEST I AND
GT40/GT44 INSTRUCTION TEST II MUST HAVE RUN IN THEIR
ENTIRETY BEFORE ATTEMPTING TO RUN THIS TEST.

3. LOADING PROCEDURE

3.1 METHOD

PROCEDURE FOR NORMAL BINARY TAPES SHOULD BE FOLLOWED.

4. STARTING PROCEDURE

4.1 STARTING ADDRESS

LOAD ADDRESS 0000
START WITH SWITCHES 7=0, 8=0 FOR AUTO SEQUENCING
THRU ALL NON-OPERATOR INTERVENTION PATTERNS.
START WITH SWITCH BIT 7=0, 8=1 FOR SWITCH REGISTER PATTERN
CONTROL (REF 4.2).
START WITH SWITCH BIT 7=1, 8=0 OR 1 FOR KEYBOARD PATTERN
CONTROL (REF 4.3).

4.2 CONTROL SWITCH SETTINGS (SWITCH REGISTER)

SWITCH REGISTER BITS 0,1,2,3 ARE USED TO SELECT EACH OF THE TESTS.

NON-OPERATOR INTERVENTION TESTS

SW 3 = 0 00 /DIRECTORY
 01 /DOT REPEATABILITY
 02 /PINCUSHION (X AND Y OFFSET ADJ.)
 03 /OCTAGONS OR SQUARES
 04 /CHARACTER SET (CHAR ADJ.)
 05 /DASH LINES AND BLINK
 06 /VECTOR LENGTH TEST < X VECTOR LENGTH ADJ.>
 07 /VECTOR LENGTH TEST < Y VECTOR LENGTH ADJ.>
 10 /PHOSPHOR TEST (HORIZ)
 11 /PHOSPHOR TEST (VERT)
 12 /INTENSITY LEVELS, SYNC AND LIGHT-PEN TEST
 13 /EDGE TEST
 14 /SHORT VECTOR AND RELATIVE POINT TEST
 15 /GRAPH PLOT INCREMENT TEST

OPERATOR INTERVENTION TESTS

 16 /LIGHT-PEN FOLLOW TEST
 17 /KEYBOARD ECHO

SW 6 = 0 SELECT SUB-PICTURE 0
SW 6 = 1 SELECT SUB-PICTURE 1 OR
 STOP DISPLAY FRAME MOTION

SW 8 = 0 EXECUTE ALL NON-OPERATOR INTERVENTION FRAMES;
SW 8 = 1 EXECUTE THE DISPLAY FRAME SPECIFIED BY SW 6-3.

4.3 CONTROL SWITCH SETTINGS (DISPLAY KEYBOARD)

ALPHA CHARACTERS 'A' THRU 'P' ARE USED TO SELECT EACH OF THE TESTS.

CHARACTER	TEST
A	DIRECTORY
B	DOT REPEATABILITY
C	PINGUSHION &X AND Y OFFSET ADJ.)
D	OCTAGONS OR SQUARES
E	CHARACTER SET (CHAR. ADJ.)
F	DASH LINES AND BLINK
G	VECTOR LENGTH TEST (X VECTOR LENGTH ADJ.)
H	VECTOR LENGTH TEST (Y VECTOR LENGTH ADJ.)
I	PHOSPHOR TEST (HORIZ)
J	PHOSPHOR TEST (VERT)
K	INTENSITY LEVELS, SYNC AND LIGHT-PEN TEST
L	EDGE TEST
M	SHORT VECTOR AND RELATIVE POINT
N	GRAPHLOT INCREMENT TEST
O	LIGHT-PEN FOLLOW TEST
P	KEYBOARD ECHO

DEPRESSING A 'RUBOUT' AFTER SELECTING A FRAME WILL LOCK ON THE SELECT FRAME.

DEPRESSING A 'CR' AFTER SELECTING A FRAME WILL SELECT SUB-PICTURE 1 OR STOP DISPLAY FRAME MOTION.

TO CONTINUE AFTER DEPRESSING A 'CR' OR 'RUBOUT' DEPRESS ANY KEY OTHER THAN 'CR' OR 'RUBOUT'.

DEPRESSING 'CONTROL C (<C>)' WHEN EXECUTING THE KEYBOARD ECHO TEST, WILL RETURN CONTROL TO THE DIRECTORY FRAME.

5. OPERATING PROCEDURE

5.1 OPERATIONAL SWITCHES

ALL OF THE TEST WILL RUN IN THEIR NORMAL MANNER WITHOUT ANY OPERATIONAL SWITCHES SELECTED, HOWEVER, SOME OF THE TESTS HAVE ADDITIONAL FEATURES AND THEY ARE SELECTED BY USING SWITCH BIT 06 OR "CR" KEYBOARD KEY.

5.1.1 PINGUSHION TEST

SW 6 = 0 DISPLAY PINGUSHION
SW 6 = 1 DISPLAY CROSSHATCH <IN-HOUSE TEST ONLY>

5.1.2 OCTAGON OR SQUARES

SW 6 = 0 DISPLAY OCTAGONS
SW 6 = 1 DISPLAY SQUARES

5.1.3 VECTOR LENGTH TEST

SW 6 = 0 SWEEP MOVEMENT
SW 6 = 1 STOP MOVEMENT

5.1.4 PHOSPHOR TEST

SW 6 = 0 SWEEP ACROSS THE SCREEN
SW 6 = 1 STOP MOVEMENT

5.1.5 INTENSITY TEST

SW 6 = 0 ENABLE SYNC 'OFF'
SW 6 = 1 ENABLE SYNC 'ON'

5.1.6 GRAPH PLOT INCREMENT TEST

SW 6 = 0 USE GRAPH PLOT X
SW 6 = 1 USE GRAPH PLOT Y

5.1.7 LIGHT PEN FOLLOW

SW 6 = 0 DISPLAY LIGHT PEN FOLLOW
SW 6 = 1 DISPLAY LIGHT PEN FIELD OF VIEW
 <IN-HOUSE TEST ONLY>

6. ERRORS

THE PROGRAM WILL ONLY HALT ON ERROR.
THE PROGRAM DOES NOT CONTAIN FACILITIES FOR THE REPORTING OF ERROR CONDITIONS.

7. RESTRICTIONS

IF USING THE SWITCH REGISTER (REF #.2) TO CONTROL THE PROGRAM, THERE WILL BE A DELAY BEFORE THE NEW TEST IS SELECTED.

8. MISCELLANEOUS

8.1 DEVICE ADDRESS PROGRAM LOCATIONS

LOCATION 1000 CONTAINS THE GT40/GT44 DEVICE ADDRESS;
LOCATION 1002 CONTAINS THE GT40/GT44 INTERRUPT VECTOR,
LOCATION 1004 CONTAINS THE GT40/GT44 INTERRUPT BR LEVEL.

9. PROGRAM DESCRIPTION

9.1 DIRECTORY

THIS TEST USES THE CHARACTER MODE TO DISPLAY A DIRECTORY OF THE TESTS THAT ARE AVAILABLE.

9.2 DOT REPEATIBILITY

THIS TEST INTENSIFIES A DOT IN EACH CORNER AND A DOT IN THE CENTER OF THE SCREEN. THIS TEST IS USED TO VERIFY DOT REPEATIBILITY.

9.3 PINCUSHION AND VECTOR CURVATURE TEST (ADJUSTMENT OF X AND Y OFFSET POTS)

THIS TEST OUTLINES THE FULL SCREEN AREA. IT IS USEFUL IN CENTERING THE VIEWING AREA IN THE DISPLAY MASK. THIS TEST ALSO DRAWS A DIAGONAL LINE FROM LOWER LEFT CORNER TO THE UPPER RIGHT AND THEN RETURNS IN THE OPPOSITE DIRECTION. A SIMILAR SEQUENCE IS REPEATED STARTING AT LOWER RIGHT CORNER TO THE UPPER LEFT CORNER AND BACK. THE PURPOSE IS TO MAKE CERTAIN THAT THE VECTORS ARE LINEAR OVER THEIR ENTIRE LENGTH. WITH PROPER LENGTH VECTORS ONLY TWO DIAGONAL LINES SHOULD BE SEEN IN THE CENTER OF THE SCREEN. DO NOT ADJUST THE VECTOR LENGTH POTS WITH THIS DISPLAY PATTERN. SINGLE LINES SHOULD BE VISIBLE AT THE TOP AND BOTTOM OF THE SCREEN. IF NOT ADJUST THE Y OFFSET POT. SINGLE LINES SHOULD BE VISIBLE AT THE RIGHT AND LEFT EDGE OF THE SCREEN IF NOT ADJUST THE X OFFSET POT.

9.4 OCTAGONS OR SQUARES

A SERIES OF DIFFERENT SIZE OCTAGONS OR SQUARES ARE DRAWN TO DEMONSTRATE THAT CLOSED FIGURES CAN BE DRAWN USING DIFFERENT VECTOR LENGTHS (7,17,37,73,177,377 AND 777). THIS TEST IS USED TO TEST THE END POINT MATCHING OF THE VECTORS.

9.5 CHARACTER SET (ADJUSTMENT OF THE CHARACTER POT'S)

TWO COMPLETE SETS OF ASCII CHARACTERS AVAILABLE FROM THE CHARACTER GENERATOR ARE DISPLAYED. THE CHARACTERS ARE DISPLAYED IN FOUR LINES OF TEXT. THE FIRST HALF OF A LINE IS IN 'NORMAL' FONT. THE SECOND HALF OF A LINE IS IN 'ITALICS' FONT.

9.6 DASH LINES AND BLINK TEST

THIS TEST IS USED TO TEST THE FOUR TYPES OF VECTOR LINES. FOUR VECTORS ARE PLOTTED USING EACH OF THE FOUR LINE REGISTER VALUES. THIS TEST ALSO ENABLES THE BLINK OPTION. THE FIRST VECTOR ON A LINE SHOULD NOT BLINK. THE SECOND VECTOR ON A LINE SHOULD BLINK.

9.7 VECTOR LENGTH TEST (ADJUSTMENT OF X AND Y VECTOR LENGTH)

A SERIES OF INCREMENTING ANGLE VECTORS ARE DRAWN FROM THE SCREEN ORIGIN TO THE OPPOSITE EDGE OF THE SCREEN. THESE VECTORS SHOULD TERMINATE ON THE LINE DRAWN AT THE VIEWING EDGE. IF THE VECTORS DO NOT END ON THE LINE, ADJUST THE APPROPRIATE VECTOR LENGTH POT.

9.8 PHOSPHOR TEST

A WIDE BAND OF INTENSIFIED VECTORS IS DISPLAYED TO ALLOW FOR VISUAL INSPECTION OF THE CRT PHOSPHOR. THIS TEST ALSO TEST FOR ANY DISTORTION IN DEFLECTION CROSS-OVER IN THE SCORE.

9.9 INTENSITY LEVEL, SYNC AND LIGHT-PEN SENSITIVITY TEST

EIGHT VECTORS ARE DRAWN USING EACH OF THE EIGHT INTENSITY LEVELS. THE INTENSITY SHOULD BE ADJUSTED SO THAT THE LEVEL 8 IS BARELY VISIBLE. THIS TEST IS ALSO USED TEST THE LIGHT PEN SENSITIVITY. ALL LINES ARE SET TO ALLOW A LIGHT PEN HIT. THEN HIT THE MESSAGE 'LIGHT PEN HIT' WILL BE DISPLAYED ON THE LINE HIT. THIS TEST IS ALSO USED TO TEST THE 'SYNC' LOGIC IF SELECTED.

9.10 EDGE SQUARES TEST

THIS TEST IS USED TO TEST FOR PROPER EDGE BLANKING AND REENTRY SETTLE TIME. THE SCREEN IS OUTLINED AND FOUR RECTANGLES ARE DRAWN AS TO EXCEED THE EDGE OF THE SCREEN. ONLY HALF OF EACH RECTANGLE SHOULD BE VISIBLE.

9.11 SHORT VECTOR AND RELATIVE POINT TEST

THIS TEST IS USED TO VERIFY PROPER DECODING OF THE SHORT VECTOR AND RELATIVE POINT. A SERIES OF INTENSIFIED VERTICAL LINES ARE PLOTTED USING SHORT VECTOR MODE. THE TEST THEN REPEATS USING RELATIVE POINT. THE RESULTS IS THAT A SINGLE HORIZONTAL LINE APPEARS TO THE RIGHT OF THE VERTICAL LINES. ALSO INCLUDED IS A RELATIVE POINT REPEATABILITY TEST. FOUR SETS OF THREE OCTAGONS EACH WILL BE DISPLAYED. THE INNER OCTAGON IS DRAWN USING SHORT VECTOR MODE WITH A DELTA X, Y OF 71 OCT. THE MIDDLE OCTAGON IS DRAWN USING RELATIVE POINT MODE WITH A DELTA X, Y OF 74 OCT. THE OUTER OCTAGON IS DRAWN USING SHORT VECTOR MODE WITH AN DELTA X, Y OF 77 OCT. THE MIDDLE OCTAGON SHOULD BE EQUAL DISTANCE FROM THE OUTER OCTAGONS AND SHOULD NOT MOVE.

9.12 GRAPHLOT INCREMENT TEST

A SERIES OF POINTS ARE PLOTTED WITH EACH POSSIBLE VALUE IN THE GRAPHLOT INCREMENT REGISTER FROM 0-777 THE RESULTING PATTERN USED SHOULD APPEAR TO BE A SERIES OF POINTS AT AN INCREASING ANGLE.

9.13 LIGHT-PEN FOLLOW TEST

IN THIS OPERATOR INTERVENTION TEST A TRACKING CROSS IS DISPLAYED. THE OPERATOR MAY MOVE ACROSS THE SCREEN WITH THE LIGHT PEN. AN X AND Y OCTAL READOUT IS ALSO DISPLAYED TO THE OPERATOR.

9.14 KEYBOARD ECHO TEST

THIS IS AN OPERATOR INTERVENTION TEST USED TO INSURE PROPER OPERATION OF THE DISPLAY KEYBOARD. WHEN A DISPLAYABLE CHARACTER KEY IS DEPRESSED THE CHARACTER IS DISPLAYED ON THE SCREEN. IN SELECTING THE SHIFT-OUT MODE, IF THE KEY DEPRESSED IS NOT A CONTROL CHARACTER, THE PROGRAM WILL TRAP TO THE SHIFT-OUT VECTOR. AN OCTAL CHARACTER VALUE READOUT IS ALSO DISPLAYED AS AN AID IN ADJUSTING THE TTY CLOCK.

.LIST

368
369
370

```

372
373
374      :ENABL  ABS:AH
375      :TITLE  GT=42/RT=44 WITH VR17 VISUAL DISPLAY TEST MAINDEC=11-DDGTG=A
376      :LIST   ME
377      :NLIST  MC,MD,CND
378      000000      R0=X0
379      000001      R1=X1
380      000002      R2=X2
381      000003      R3=X3
382      000004      R4=X4
383      000005      R5=X5
384      000006      SP=X6
385      000007      PC=X7
386      104000      SCOPE=ENT
387      000500      STKPTR=500
388      177570      DISPLAY=177570      11/45 LIGHT DISPLAY REGISTER
389
390      10=776 IS FILLED WITH 102, HALT
391      :LIST
392
401      000024      000024
402      000024      001250
403      000026      000340
404
405      000030      000030
406      000030      001100
407      000032      000340
408
409      :WORD  LDWPHR
410      340
411
412      :WORD  SCOPEA      JEMT RETURN
413      340

```

```

410
411      000000      000000
412      000200      000137      001356      :JMP  START  IDISPLAY TEST
413
414      001000      001000
415      001000      172000      GSADD: 172000      IDISPLAY STARTING ADDRESS
416      001002      000320      GSVC1: 320      IDISPLAY INTERRUPT VECTOR STARTING ADDRESS
417      001004      000200      GSRL: 200      IDISPLAY BR LEVEL
418
419      001006      000000      ICNT: 0
420      001010      177776      PSW: 177776
421      001012      177560      TKS: 177560
422      001014      177562      TK5: 177562
423      001016      012470      DBUF1: BUFFER      IFIRST WORD IN THE DISPLAY BUFFER
424      001020      012472      DBUF2: BUFFER+2      ISECOND WORD
425      001022      012474      DBUF3: BUFFER+4      ITHIRD WORD
426      001024      012476      DBUF4: BUFFER+6      IFOURTH WORD
427      001026      012500      DBUF5: BUFFER+10      IFIFTH WORD
428      001028      012502      DBUF6: BUFFER+12
429      001032      000000      DSAVE1: 0      ITEMP REG,
430      001034      000000      DSAVE2: 0
431      001036      000000      DSAVE3: 0
432      001040      000000      DSAVE4: 0
433      001042      000000      HOLD1: 0
434      001044      000000      TSAVE1: 0
435      001046      000000      CNTR1: 0
436      001050      000000      CHANGE1: 0
437      001052      000000      LDKRB: 0
438
439
440
441
442
443
444
445
446      001054      172000      DPCI: 172000      IDISPLAY PROGRAM COUNTER
447      001056      172002      DSRI: 172002      IDISPLAY STATUS REGISTER
448      001060      172004      XPSI: 172004      IDISPLAY X AXIS REGISTER
449      001062      172006      YPSI: 172006      IDISPLAY Y AXIS REGISTER
450
451      001064      000320      DDONE1: 320      IDISPLAY INTERRUPT VECTOR FOR STOP
452      001066      000322      DDONE2: 322
453
454      001070      000324      LPVCT1: 324      IDISPLAY INTERRUPT VECTOR FOR LIGHT-PEN
455      001072      000326      LPVCT2: 326
456
457      001074      000330      TIMEVT1: 330      IDISPLAY INTERRUPT VECTOR FOR TIME-OUT OR SHIFT-OUT
458      001076      000332      TIMEVT2: 332
459

```

```

461                                MONITOR ROUTINE
462
463 001120 005737 002046 SCOPEA1 TST KRBQ      ;TEST IF SW OR "KRB"
464 001144 001014          BNF SCOPEF      ;BR IF "KRB"
465 001106 005037 005556 CLR SWITCH    ;CLEAR "SWITCH"
466 001122 032737 000100 BIT #10000000 ;TEST FOR "HOLD/STOP SWITCH"
467 001120 001402          BEQ SCOPEE      ;BR IF CLEARED
468 001122 005137 005556 COM SWITCH    ;SET SWITCH
469 001126 032737 000400 BIT #40000000 ;TEST BIT 0
470 001134 001010          BNE SCOPEB      ;
471 001136 005737 001042 SCOPEF1 TST WOLG      ;TEST FOR "HOLD/STOP"
472 001142 001012          BNF SCOPEO      ;BR IF SET
473 001144 000240          NOP
474 001146 004737 001536 JSR PC,SETUP      ;RESET HOUSEKEEPING
475 001152 000240          NOP
476 001154 000002          RTI
477 001156 013704 177570 SCOPEB1 MOV #0DISPLAY,R4 ;READ SWITCHES
478 001162 042704 177760 SCOPEC1 BIC #17760,R4 ;MASK TO BITS 4-15
479 001166 000304          ASL R4           ;MOVE LEFT
480 001170 012706 000500 SCOPEO1 MOV #STKPTR,SP ;RESET STACK
481 001174 000240          NOP
482 001176 004737 001536 JSR PC,SETUP      ;RESET HOUSEKEEPING
483 001202 000240          NOP
484 001204 000174 001210 JMP PO1PTC(R4) ;JUMP TO THAT TEST
485
486 001230 002052 DISPTC1 FILE0+2 ;DIRECTORY
487 001232 002064 FILE1+2 ;DOT REPEATABILITY
488 001234 002076 FILE2+2 ;PINCUSHION
489 001236 002342 FILE3+2 ;OCTAGONS OR SQUARES
490 001220 002416 FILE4+2 ;CHARACTER SET
491 001222 003026 FILE5+2 ;DASH LINES AND BLINK
492 001224 003040 FILE6+2 ;X VECTOR LENGTH
493 001226 003172 FILE7+2 ;Y VECTOR LENGTH
494 001230 003324 FILE10+2 ;X PHOSPHOR TEST
495 001232 003400 FILE11+2 ;Y PHOSPHOR TEST
496 001234 003454 FILE12+2 ;INTENSITY LEVEL AND LIGHTPEN
497 001236 003616 FILE13+2 ;CODE SQUARES
498 001240 003630 FILE14+2 ;SHORT VECTOR RELATIVE POINT TEST
499 001242 004110 FILE15+2 ;GRAPHPLOT TEST
500 001244 004344 FILE16+2 ;LIGHT-PEN FOLLOW
501 001246 005054 FILE17+2 ;KEY BOARD ECHO
502

```

```

504
505
506 001250 010044 LONPHR1 MOV R0,*(SP)
507 001252 010146      MOV R1,*(SP)
508 001254 010246      MOV R2,*(SP)
509 001256 010346      MOV R3,*(SP)
510 001260 010446      MOV R4,*(SP)
511 001262 010546      MOV R5,*(SP)
512 001264 010637 001300      MOV SP,LONSV
513 001270 012737 001302 000024      MOV #HIGPHR1,#24
514 001276 000000      HALT
515
516 001300 000000 LONSV1 0
517
518 001302 013704 001300 HIGPHR1 MOV LONSV,SP
519 001306 012605      MOV (SP)+,R3
520 001310 012604      MOV (SP)+,R4
521 001312 012603      MOV (SP)+,R3
522 001314 012602      MOV (SP)+,R2
523 001316 012601      MOV (SP)+,R1
524 001320 012600      MOV (SP)+,R0
525 001322 012737 001250 000024      MOV #LONPHR1,#24
526 001330 012706 000500      MOV #STKPTR,SP
527 001334 000240      NOP
528 001336 000240      NOP
529 001340 000240      NOP
530 001342 000000      HALT
531 001344 000240      NOP
532 001346 000240      NOP
533 001350 000240      NOP
534 001352 000137 001170      JMP SCOPEO

```

```

563      001536  012737  000062  000060  SETUP1  MOV      #02;#60      IRESET KRB VECTOR
564      001544  012737  000000  000062  MOV      #0;#62
565      001552  042777  000100  177232  BIC      #100;#THS      ICLEAR INT ENABLE
566      001560  000337  000246  CLR      KRB0
567      001564  032737  000246  177570  BIT      #200;#DISPLAY  ITEST FOR "KRB" CONTROL
568      001572  001413  SEC      SETUPA        ISR IF NOT
569      001574  001413  000246  COM      KRB0          ISET "KRB" CONTRQL
570      001580  012737  001700  000060  MOV      KRB1;#60      ISET UP "KRB" INT
571      001604  012737  000340  000062  MOV      #340;#62
572      001616  012737  000340  000062  MOV      #100;#THS      IENABLE "KRB" INT
573      001614  052777  000100  177170  BIS      #100;#THS      ISET UP GT OONE VECTOR
574      001632  012777  001664  177234  SETUPA# MOV      #SETUP1;#DOONE  IRESET LIGHT-PEN VECTOR
575      001638  012777  000340  177230  MOV      #340;#DOONE#
576      001636  013777  001072  177224  LPMV1;1,9LPMV1        IRESET TIME-OUT/SHIFT OUT VECTOR
577      001644  000077  177222  CLR      #LPMV1;1
578      001650  013777  001076  177246  MOV      TIMEV1;#TIMEV1  IRESET TIME-OUT/SHIFT OUT VECTOR
579      001656  000077  177244  CLR      #TIMEV1;1
580      001662  000207  RTS
581
582      001664  000777  177106  SETUPB# TST      #000#      ITEST FOR STOP
583      001670  100401  BMI
584      001672  000000  HALT      .+4
585
586      001674  000002  RTI
587      001676  000000  HALT
588

```

```

589
590
591 001730 117937 177110 001044 RETBI MOVB      #KYTTSAVE      IREAD THE CHARACTER
592 000178 042737 177600 001044      BIC      #177600,TSAVE      MASK TO 7 BITS
593 000171 022787 000015 001044      CMP      #15,TSAVE      TEST FOR "CR"
594 000172 000440      BEQ      KYT3      BR IF
595 000172 000939 000556      CLR      SWITCH      CLEAR "SWITCH"
596 000173 162739 000101 001044      SUB      #103,TSAVE      MAKE 0-77
597 000173 100426      KYT5I BMI      KYT3      IKA
598 000174 022737 000017 001044      CMP      #17,TSAVE
599 000174 100412      8MI      KYT2      I>P
600 000173 013704      MOV      TSAVE,R4
601 000174 012737 177777 001030      MOV      #=1,CHANGE
602 000170 000937 000556      CLR      SWITCH
603 000176 000939 001042      CLR      HOLD
604 000172 000002      RTI
605 000174 022737 000076 001044 KYT2I CMP      #76,TSAVE      IEXIT
606 000002 001015      BNE      KYT3
607 000004 012737 177777 001042      MOV      #=1,HOLD      RUBOUT
608 000012 000002      RTI      IEXIT
609 000014 000937 001042      KYT1I CLR      HOLD
610 000002 000002      RTI
611 000022 000000      HALT      IFATAL ERROR RTI FAILED
612
613 000204 012737 177777 000556      KYT3I MOV      #=1,SWITCH
614 000202 000002      RTI
615 000204 000000      HALT      IFATAL ERROR, RTI FAILED
616
617 000206 162737 000040 001044 KYT4I SUB      #40,TSAVE      ICONVERT LC TO UC
618 000204 000734      8R      KYT3
619 000206 000000      KRBU0
620

```

```

685 .LIST
689 IEXECUTE DIRECTORY FRAME
690
691 002090 104000
692 002092 004537 005412
693 002096 001000
694 0020A0 005560
695
696 IEXECUTE DOT REPEATABILITY FRAME
697
698 002062 104000
699 002064 004537 005412
700 002070 100000
701 002072 007140
702
703 IEXECUTE PINCUSHION FRAME
704
705 002074 104000
706 002076 012700 012470
707 002102 004737 002252
708 002106 012701 000020
709 002112 012720 040000
710 002116 012720 001777
711 002122 012720 000100
712 002126 012720 021777
713 002132 005301
714 002134 001366
715 002136 012720 000001
716 002142 012720 000000
717 002146 012720 040000
718 002152 012720 001777
719 002156 004737 002252
720 002162 012701 000020
721 002166 012720 041777
722 002172 012720 000000
723 002176 012720 021777
724 002202 012720 000100
725 002206 005301
726 002210 001366
727 002212 012720 000000
728 002216 012720 000001
729 002222 012720 041777
730 002226 012720 000000
731 002232 012720 173400
732 002236 012720 100000
733 002242 012710 012470
734 002246 000137 002274
735
736 002252 012720 117000
737 002256 012720 000020
738 002262 012720 000000
739 002266 012720 110000
740 002272 000287

```

FILE0: SCOPE JSR 5,MSG IEXIT TO DISPLAY A FRAME
1000 FRME0 IUSING THE DIR. FRAME

FILE1: SCOPE JSR 5,MSG IEXIT TO DISPLAY A FRAME
100000 FRME1 IUSING THE DOT REPEAT FRAME

FILE2: SCOPE
MOV #BUFFER,R0 ILOAD START ADDRESS
JSR PC,SETPNT ILOAD 0,0 ORGIN
MOV #20,R1 ISETUP COUNT
1ST MOV #INTX,(R0)+ ILOAD INT LINE
MOV #MAXY,(R0)+ I MAX Y
MOV #00,(R0)+ ILOAD DELTA X
MOV #MINUS*MAXV,(R0)+ ILOAD Y MAX Y
DEC R1 IF FINISHED ?
BNE 15 IBR IF NOT
MOV #MINUS*1,(R0)+ IGO BACK 1 UNIT
MOV #0,(R0)+
MOV #INTX,(R0)+
MOV #MAXY,(R0)+
JSR PC,SETPNT ILOAD LAST LINE
MOV #MAXY*100,R1 ISET ORGIN
2ST MOV #INTX*MAXX,(R0)+ ISETUP COUNT
MOV #0,(R0)+ ILOAD DELTA X MAX
MOV #MINUS*MAXX,(R0)+ ILOAD DELTA Y # 0
MOV #00,(R0)+ IRETRAD
DEC R1 ILOAD DELTA Y OF 100
BNE 25 IF FINISHED ?
MOV #0,(R0)+ IBR IF NOT
MOV #MINUS*1,(R0)+
MOV #INTX*MAXX,(R0)+ ILOAD LAST LINE
MOV #0,(R0)+
MOV #OSJOP,(R0)+ ILOAD STOP
MOV #0JRP,(R0)+ ILOAD JUMP
MOV #BUFFER,(R0)
JMP FILE2A

SETPNT: MOV #POINT*INTX,(R0)+ ILOAD POINT
MOV #0,(R0)+ I AT X
MOV #0,(R0)+ I AT Y
MOV #LONGV,(R0)+ I LONG VECTOR
RTS PC IEXIT

```

742 002274 012737 004000 001046
743 002302 005737 005556
744 002306 001405
745 002310 004537 005412
746 002314 000001
747 002316 012470
748 002320 000404
749
750
751 002322 004537 005412
752 002326 000001
753 002330 007250
754 002332 005337 001046
755 002336 001361
756
757 IEXECUTE OCTAGONS OR SQUARES
758
759 002340 104000
760 002342 012737 014000 001046
761 002350 005737 005556
762 002354 001010
763 002356 004537 005412
764 002362 000001
765 002364 007334
766 002366 005337 001046
767 002372 001366
768 002374 000407
769
770 002376 004537 005412
771 002402 000001
772 002404 007724
773 002406 005337 001046
774 002412 001356

```

FILE2A: MOV #4000,CNTR ILOAD COUNTER
FILE2B: TST SWITCH ITEST SWITCH
BEQ FILE2C IBR IF SUBTEST NOT SELECTED
JSR R5,MSG IEXIT TO DISPLAY FRAME
1
BUFFER IUSING THE CROSS MATCH PATTERN
BR FILE2D IBR

FILE2C: JSR R5,MSG IEXIT TO DISPLAY FRAME
1
FRME2 IUSING THE OFFSET PATTERN

FILE2D: DEC CNTR IF FINISHED ?
BNE FILE2B IBR IF NOT

FILE3: SCOPE
MOV #14000,CNTR ISET UP A COUNTER
FILE3A: TST SWITCH
BNE FILE3B IBRANCH IF SUB+TEST
JSR 5,MSG IDISPLAY TEST
1
FRME3 IFRAME # 3
DEC CNTR IDECREMENT COUNTER
BNE FILE3A IBRANCH IF NOT COMPLETE
BR FILE4 IEXIT TO NEXT TEST

FILE3B: JSR 5,MSG IDISPLAY TEST
1
FRME3A IFRAME # 3A
DEC CNTR IDECREMENT COUNTER
BNE FILE3A IBRANCH IF NOT COMPLETE

```

776
777
778      ;DISPLAY FILE
779      ;CHARACTER AND ITALICS TEST
780      ;SET UP THE BUFFER FOR THIS TEST
781      FILE4: SCOPE
782      MOV     $BUFFER,R0
783      MOV     #STAT$SIZE0,(R0)+
784      MOV     #STAT$ITAL$SYNOFF$GREEN,(R0)+
785      MOV     #POINT$INT4$LOFF$BLKOFF$LINE0,(R0)+      ;LCAO POINT MPDE
786      MOV     #0,(R0)+
787      MOV     #MAXY=77,(R0)+
788      MOV     #CHAR,(R0)+
789      MOV     #17,(R0)+
790      MOV     #17,(R0)+
791      MOV     #100,$STCHAR      ;LCAO INITIAL CHAR,
792      JSR     PC,LOADBF
793      MOV     #14,$STCHAR      ;LCAO INITIAL LC CHAR
794      JSR     PC,LOADBF
795      MOV     #40,$STCHAR      ;LCAO LINE
796      JSR     PC,LOADBF
797      MOV     #STAT$ITAL0,(R0)+      ;LCAO NUMBERS AND PUNCT
798      JSR     PC,LOADBF
799      MOV     #STAT$ITAL1,(R0)+      ;LCAO LINE
800      JSR     PC,LOADBF
801      MOV     #STAT$ITAL1,(R0)+      ;LCAO NORMAL FONT
802      JSR     PC,LOADBF
803      MOV     #0,$TOP,(R0)+      ;LCAO SPECIAL CHARS
804      JSR     PC,LOADBF
805      MOV     #0,$TOP,(R0)+      ;LCAO INSERT SPACES
806      JSR     PC,LOADBF
807      MOV     #0,$TOP,(R0)+      ;LCAO ITALICS FONT
808      JSR     PC,LOADBF
809      MOV     #0,$TOP,(R0)+      ;LCAO SPECIAL
810      JSR     PC,LOADBF
811      MOV     #0,$TOP,(R0)+      ;LCAO BSYOP
812      JUMP    FILE4
813
814      LOADSP: MOV     #16,(R0)+
815      MOV     #0,$2
816      MOV     #37,$3
817      MOV     #R2,(R0)+
818      INC     R2
819      CMP     #17,R2
820      BEQ     2$
821      DEC     R3
822      BNE     1$
823      MOV     #20,$17,(R0)+
824      RTS     PC
825
826      ;LCAO INITIAL SHIFT OUT CHAR
827      ;LCAO COUNT
828      ;LCAO CHAR
829
830      ;TEST FOR 81
831      ;IF 81 "17"
832      ;IF FINISHED ?
833      ;IF 81 NOT
834      ;LCAO SHIFT=IN SPACE
835      ;EXIT
836
837      ;LCAO NORMAL FONT
838      ;LCAO STARTING CHAR
839      ;LCAO THE CHARACTERS
840      ;LCAO INSERT SPACES
841      ;LCAO ITALICS FONT
842      ;LCAO STARTING CHARACTER
843      ;LCAO THE CHARACTERS
844      ;LCAO INSERT CR/LF
845      ;EXIT
846
847      ;LCAO CR/LF
848      ;EXIT
849
850      ;LCAO CR/LF
851      ;EXIT
852
853      ;LCAO CR/LF
854      ;EXIT
855
856      ;LCAO CR/LF
857      ;EXIT
858
859      ;LCAO CR/LF
860      ;EXIT
861
862      ;LCAO CR/LF
863      ;EXIT
864
865      ;LCAO CR/LF
866      ;EXIT
867
868      ;LCAO CR/LF
869      ;EXIT
870
871      ;LCAO CR/LF
872      ;EXIT
873
874      ;LCAO CR/LF
875      ;EXIT
876
877      ;LCAO CR/LF
878      ;EXIT
879
880      ;LCAO CR/LF
881      ;EXIT
882
883      ;LCAO CR/LF
884      ;EXIT
885
886      ;LCAO CR/LF
887      ;EXIT
888
889      ;LCAO CR/LF
890      ;EXIT
891
892      ;LCAO CR/LF
893      ;EXIT
894
895      ;LCAO CR/LF
896      ;EXIT
897
898      ;LCAO CR/LF
899      ;EXIT
900
901      ;LCAO CR/LF
902      ;EXIT
903
904      ;LCAO CR/LF
905      ;EXIT
906
907      ;LCAO CR/LF
908      ;EXIT
909
910      ;LCAO CR/LF
911      ;EXIT
912
913      ;LCAO CR/LF
914      ;EXIT
915
916      ;LCAO CR/LF
917      ;EXIT
918
919      ;LCAO CR/LF
920      ;EXIT
921
922      ;LCAO CR/LF
923      ;EXIT
924
925      ;LCAO CR/LF
926      ;EXIT
927
928      ;LCAO CR/LF
929      ;EXIT
930
931      ;LCAO CR/LF
932      ;EXIT
933
934      ;LCAO CR/LF
935      ;EXIT
936
937      ;LCAO CR/LF
938      ;EXIT
939
940      ;LCAO CR/LF
941      ;EXIT
942
943      ;LCAO CR/LF
944      ;EXIT
945
946      ;LCAO CR/LF
947      ;EXIT
948
949      ;LCAO CR/LF
950      ;EXIT
951
952      ;LCAO CR/LF
953      ;EXIT
954
955      ;LCAO CR/LF
956      ;EXIT
957
958      ;LCAO CR/LF
959      ;EXIT
960
961      ;LCAO CR/LF
962      ;EXIT
963
964      ;LCAO CR/LF
965      ;EXIT
966
967      ;LCAO CR/LF
968      ;EXIT
969
970      ;LCAO CR/LF
971      ;EXIT
972
973      ;LCAO CR/LF
974      ;EXIT
975
976      ;LCAO CR/LF
977      ;EXIT
978
979      ;LCAO CR/LF
980      ;EXIT
981
982      ;LCAO CR/LF
983      ;EXIT
984
985      ;LCAO CR/LF
986      ;EXIT
987
988      ;LCAO CR/LF
989      ;EXIT
990
991      ;LCAO CR/LF
992      ;EXIT
993
994      ;LCAO CR/LF
995      ;EXIT
996
997      ;LCAO CR/LF
998      ;EXIT
999
1000     ;LCAO CR/LF
1001     ;EXIT

```

```

830
831      CR/LF: MOV     #13,(R0)+
832      MOV     #12,(R0)+
833      MOV     #12,(R0)+
834      MOV     #12,(R0)+
835      RTS     PC
836
837      ;EXIT
838
839      ;LCAO CR/LF
840      ;EXIT
841
842      ;LCAO CR/LF
843      ;EXIT
844
845      ;LCAO CR/LF
846      ;EXIT
847
848      ;LCAO CR/LF
849      ;EXIT
850
851      ;LCAO CR/LF
852      ;EXIT
853
854      ;LCAO CR/LF
855      ;EXIT
856
857      ;LCAO CR/LF
858      ;EXIT
859
860      ;LCAO CR/LF
861      ;EXIT
862
863      ;LCAO CR/LF
864      ;EXIT
865
866      ;LCAO CR/LF
867      ;EXIT
868
869      ;LCAO CR/LF
870      ;EXIT
871
872      ;LCAO CR/LF
873      ;EXIT
874
875      ;LCAO CR/LF
876      ;EXIT
877
878      ;LCAO CR/LF
879      ;EXIT
880
881      ;LCAO CR/LF
882      ;EXIT
883
884      ;LCAO CR/LF
885      ;EXIT
886
887      ;LCAO CR/LF
888      ;EXIT
889
890      ;LCAO CR/LF
891      ;EXIT
892
893      ;LCAO CR/LF
894      ;EXIT
895
896      ;LCAO CR/LF
897      ;EXIT
898
899      ;LCAO CR/LF
900      ;EXIT
901
902      ;LCAO CR/LF
903      ;EXIT
904
905      ;LCAO CR/LF
906      ;EXIT
907
908      ;LCAO CR/LF
909      ;EXIT
910
911      ;LCAO CR/LF
912      ;EXIT
913
914      ;LCAO CR/LF
915      ;EXIT
916
917      ;LCAO CR/LF
918      ;EXIT
919
920      ;LCAO CR/LF
921      ;EXIT
922
923      ;LCAO CR/LF
924      ;EXIT
925
926      ;LCAO CR/LF
927      ;EXIT
928
929      ;LCAO CR/LF
930      ;EXIT
931
932      ;LCAO CR/LF
933      ;EXIT
934
935      ;LCAO CR/LF
936      ;EXIT
937
938      ;LCAO CR/LF
939      ;EXIT
940
941      ;LCAO CR/LF
942      ;EXIT
943
944      ;LCAO CR/LF
945      ;EXIT
946
947      ;LCAO CR/LF
948      ;EXIT
949
950      ;LCAO CR/LF
951      ;EXIT
952
953      ;LCAO CR/LF
954      ;EXIT
955
956      ;LCAO CR/LF
957      ;EXIT
958
959      ;LCAO CR/LF
960      ;EXIT
961
962      ;LCAO CR/LF
963      ;EXIT
964
965      ;LCAO CR/LF
966      ;EXIT
967
968      ;LCAO CR/LF
969      ;EXIT
970
971      ;LCAO CR/LF
972      ;EXIT
973
974      ;LCAO CR/LF
975      ;EXIT
976
977      ;LCAO CR/LF
978      ;EXIT
979
980      ;LCAO CR/LF
981      ;EXIT
982
983      ;LCAO CR/LF
984      ;EXIT
985
986      ;LCAO CR/LF
987      ;EXIT
988
989      ;LCAO CR/LF
990      ;EXIT
991
992      ;LCAO CR/LF
993      ;EXIT
994
995      ;LCAO CR/LF
996      ;EXIT
997
998      ;LCAO CR/LF
999      ;EXIT
1000     ;LCAO CR/LF

```

```

876
877
878
879 003036 104000 FILE6: SCOPE
880 003040 012737 041777 010472 MOV #10,DELT6 ISET UP VERTICAL HEIGHT
881 003046 012737 000010 001036 MOV #10,DSAVE2 ISET UP TIER
882 003054 012737 000000 001034 MOV #0,DSAVE1
883 003062 012737 000040 001046 LOOPA1 MOV #40,CNTR ISET UP EXECUTION COUNT
884 003070 012737 000000 001032 LOOPA1 MOV #MAXY*1/10,DSAVE ISET UP
885 003076 013737 001034 010474 MOV DSAVE1,DELT6
886 003104 004537 005412 JSR 5,MESG IEXIT TO DISPLAY FRAME
887 003110 000001 1
888 003112 010426 FRME6 IVECTOR LENGTH FRAME
889 003114 004537 005412 LOOPA2 JSR 5,MESG IEXIT TO DISPLAY FRAME
890 003120 000001 1
891 003122 010462 FRME6A IVECTOR LENGTH FRAME
892 003124 062737 000010 010474 ADD #10,DELT6 IUPDATE ANGLE
893 003132 005337 001032 DEC DSAVE IFINISHED ALL THE ANGLES
894 003136 001366 BNE LOOPA2 IBR IF NOT
895 003140 005337 001046 LOOPA3 DEC CNTR IDONE COUNT?
896 003144 001351 BNE LOOPA1 IBR IF NOT
897 003146 000240 NOP
898 003150 005737 TST SWITCH ITEST SWITCH
899 003154 001342 BNE LOOPA IBR IF HALT MOTION
900 003156 005237 INC DSAVE1 IUPDATE INITIAL ANGLE
901 003160 005337 001034 DEC DSAVE2 IFINISHED ALL?
902 003166 001335 BNE LOOPA IBR IF NOT
903
904
905
906 003170 104000 FILE7: SCOPE
907 003172 012737 040000 001034 MOV #10,DSAVE1 ISETUP INITIAL X
908 003200 012737 001777 010474 MOV #MAXY,DELT6 ISETUP INITIAL Y
909 003206 012737 000010 001036 MOV #10,DSAVE2 ISETUP EXECUTION COUNT
910 003214 012737 000040 001046 LOOPB1 MOV #40,CNTR ISETUP DELAY
911 003222 012737 000000 001032 LOOPB1 MOV #200,DSAVE
912 003230 013737 001034 010472 MOV DSAVE1,DELT6 IEXIT TO DISPLAY FRAME
913 003236 004537 005412 JSR 5,MESG IVECTOR LENGTH TEST FRAME
914 003242 000001 1
915 003244 010426 FRME6 IEXIT TO DISPLAY FRAME
916 003246 004537 005412 LOOPB2 JSR 5,MESG IEXIT TO DISPLAY FRAME
917 003252 000001 1
918 003254 010462 FRME6A IVECTOR LENGTH FRAME
919 003256 062737 000010 010472 ADD #10,DELT6 IUPDATE ANGLE
920 003264 005337 001032 DEC DSAVE IFINISHED ALL THE ANGLES
921 003270 001366 BNE LOOPB2 IBR IF NOT
922 003272 005337 001046 LOOPB3 DEC CNTR IDONE COUNT?
923 003276 001351 BNE LOOPB1 IBR IF NOT
924 003300 000240 NOP
925 003302 005737 TST SWITCH ITEST SWITCH
926 003306 001342 BNE LOOPB IBR IF HALT MOTION
927 003310 005237 INC DSAVE1 IUPDATE INITIAL ANGLE
928 003314 005337 001034 DEC DSAVE2 IFINISHED ALL?
929 003320 001335 BNE LOOPB IBR IF NOT

```

```

931
932
933
934 003322 104000 FILE10: SCOPE
935 003324 005037 010506 CLR DELTY7
936 003330 004537 005412 JSR 5,MESG IEXIT TO DISPLAY A FRAME
937 003334 000000 50
938 003336 010504 FRME10
939 003340 004537 005412 JSR 5,MESG IUSING THE HORIZ FRAME
940 003344 000001 1 IEXIT TO DISPLAY A FRAME
941 003346 010504 FRM10
942 003350 000240 NOP IUSING THE PERIMETER BOX
943 003352 005737 005556 TST SWITCH ITEST THE "SWITCH"
944 003356 001364 BNE 07A IBR IF FREEZE THE MOVEMENT
945 003360 062737 000001 010506 D7C1 ADD #1,DELT7 IUPDATE THE X ORIGIN
946 003366 022737 002000 010506 CMP #200,DELT7 ITEST IF THE END
947 003374 001355 BNE 07A IBR IF NOT
948
949
950
951 003376 104000 FILE11: SCOPE
952 003400 005037 010550 CLR DELTY7
953 003404 004537 005412 JSR 5,MESG IEXIT TO DISPLAY A FRAME
954 003410 000000 50
955 003412 010544 FRME11
956 003414 004537 005412 JSR 5,MESG IUSING THE VERT FRAME
957 003420 000001 1 IEXIT TO DISPLAY A FRAME
958 003422 010504 FRM10
959 003424 000240 NOP IUSING THE PERIMETER BOX
960 003426 005737 005556 TST SWITCH ITEST THE "SWITCH"
961 003432 001364 BNE 07D IBR IF FREEZE THE MOVEMENT
962 003434 062737 000001 010550 D7F1 ADD #1,DELT7 IUPDATE THE Y ORIGIN
963 003442 022737 002000 010550 CMP #MAXY*1,DELT7 ITEST IF THE END
964 003446 001355 BNE 07D IBR IF NOT
965

```

```

967
968
969
970 003492 014000 FILE12) SCOPE
971 003494 012777 003550 175406 MOV #RETL,0LPVCT ISET UP LIGHT-PEN VECTOR
972 003492 013777 001004 175402 MOV GSRL,0LPVCT1 ISET UP BR LEVEL
973 003470 012737 004000 001032 MOV #4000,0SAVE ISET UP A EXECUTION COUNT
974 003476 005737 005556 FILE12A) TST SWITCH ITEST THE "SWITCH"
975 003502 001004 BNE FILE12B IBR IF SET "SYNC"
976 003504 042737 000004 010630 BIC #4,SYN12 IENSURE CLEAR "SYNC"
977 003532 000403 BR FILE12C IBY PASS
978 003534 052737 000004 010630 FILE12B) BIS #4,SYN12 ISET THE "SYNC"
979 003532 004537 005412 FILE12C) JSR 5,MESG IEXIT TO DISPLAY FRAME
980 003526 000001 1 FRME12
981 003530 010442 DEC IUSING THE "INTENSITY" FRAME
982 003532 005337 001032 DEC IFINISHED?
983 003536 001423 BEO FILE12D IYES, EXIT
984 003540 012737 173400 011250 MOV #0STOP,RAYLPA INO, RESET MESSAGE
985 003546 000053 BR FILE12A IBR BACK
986 003550 012737 164000 011250 RETLP) MOV #0NDP,RAYLPA ILIGHT-PEN HIT
987 003536 017737 175300 011262 MOV #YPOS,LPPNT IREAD Y POSITION
988 003504 042737 176000 011262 BIC #17000,LPPNT IMASK THE BITS
989 003572 022626 CMP (SP),*(SP)* IPOP THE STACK
990 003574 012777 000001 175252 MOV #1,0DPC ISINGLE STEP THE DISPLAY
991 003602 000137 005430 JMP MSGA IJUMP TO WAIT
992 003606 013777 001072 175254 FILE12D) MOV LPVCT1,0LPVCT IRESET THE LIGHT-PEN VECTOR
993
994
995
996
997 003614 104000 FILE13) SCOPE
998 003616 004537 005412 JSR 5,MESG IEXIT TO DISPLAY FRAME
999 003622 010000 10000 FRME13
1000 003624 011312 IUSING THE "EDGE" FRAME

```

```

1002
1003
1004
1005 003626 104000 FILE14) SCOPE
1006 003630 012700 012470 MOV #BUFFER,0 ISET UP 0
1007 003634 012720 114000 MOV #PCINT,(0)* ISET UP INITIAL
1008 003640 012720 000240 MOV #240,(0)* IX POSITION
1009 003644 012720 001000 MOV #MAXY*1/2,(0)* IY POSITION
1010 003630 012720 107004 MOV #SHORTV*INT4*LINE,(0)* ILOAD "SHORT VECTOR"
1011 003634 004737 003706 JSR PC,LOADVT ILOAD THE DISPLAY PATTERN
1012 003660 012720 130000 MOV #RELATV,(0)* ILOAD "RELATIVE POINT"
1013 003664 004737 003706 JSR PC,LOADVT ILOAD THE DISPLAY PATTERN
1014 003670 012720 173400 MOV #0STOP,(0)* ILOAD "DISPLAY STOP"
1015 003674 012720 160000 MOV #DJMP,(0)* ILOAD "DISPLAY JUMP"
1016 003700 012720 012470 MOV #BUFFER,(0)* ITO THE BUFFER ADDRESS
1017 003704 000413 BR FILE14A IBR TO THE FRAME
1018
1019 003736 012737 000024 001046 LDAOVT) MOV #24,CNTR ILOAD A COUNTER
1020 003714 012720 040077 LADVT) MOV #INTX*77,(0)* ILOAD A DELTA Y
1021 003720 012720 004177 MOV #4177,(0)* ILOAD A DELTA X,Y
1022 003724 005337 001046 DEC CNTR IFINISHED?
1023 003730 001371 BNE LADVT IBR IF NOT
1024 003732 000207 RTS PC IEXIT
1025
1026 003734 012737 004000 004104 FILE14A) MOV #4000,105 ILOAD COUNTER
1027 003742 012737 000200 011572 ISF MOV #200,FRM14A ILOAD FIRST OCTAGON
1028 003750 012737 000200 011574 MOV #200,FRM14B
1029 003756 004537 005412 JSR R5,MESG IDISPLAY OCT.
1030 003762 000001 1
1031 003764 011566 1 FRME14
1032 003766 012737 001400 011572 MOV #140,FRM14A ILOAD SECOND OCTAGON
1033 003774 012737 000200 011574 MOV #200,FRM14B
1034 004002 004537 005412 JSR R5,MESG IDISPLAY 2ND OCT.
1035 004006 000001 1
1036 004010 011566 1 FRME14
1037 004012 012737 001400 011572 MOV #140,FRM14A ILOAD THIR OCTAGON
1038 004020 012737 001400 011574 MOV #MAXY*377,FRM14B
1039 004026 004537 005412 JSR R5,MESG
1040 004032 000001 1
1041 004034 011566 1 FRME14
1042 004036 012737 000200 011572 MOV #200,FRM14A ILOAD FOURTH OCTAGON
1043 004044 012737 001400 011574 MOV #MAXY*377,FRM14B
1044 004052 004537 005412 JSR R5,MESG IDISPLAY 4TH OCT.
1045 004056 000001 1
1046 004060 011566 1 FRME14
1047 004062 004537 005412 JSR R5,MESG IDISPLAY BAR
1048 004066 000001 1
1049 004070 012470 BUFFER
1050 004072 005337 004104 DEC 105 IFINISHED ?
1051 004076 001321 BNE 15 IBR IF NOT
1052 004100 000137 004106 JMP FILE15 INEXT TEST
1053 004104 000000 105) 0
1054

```

```

1056
1057
1058
1059 004106 104000
1060 004110 012700 012470
1061 004114 012700 117600
1062 004120 012700 000000
1063 004124 012700 000000
1064 004130 012700 170052
1065 004134 012700 174100
1066 004140 012700 120000
1067 004144 012700 000040
1068 004150 012737 000000 001032
1069 004156 000403
1070 004100 062737 000020 001032
1071 004166 013720 001032
1072 004172 005305
1073 004174 001371
1074 004176 012700 173400
1075 004202 012700 160000
1076 004206 012700 012470
1077 004212 012737 000200 001032
1078 004230 002777 004000 174602
1079 004246 005737 005556
1080 004232 001403
1081 004234 002777 004000 174566
1082 004242 004537 005412
1083 004246 000001
1084 004250 012470
1085 004292 002777 000001 174546
1086 004260 002777 174200 174540
1087 004266 001346
1088 004270 012777 174100 174530
1089 004276 005337 001032
1090 004302 001346
1091
1092 004304 013700 000042
1093 004310 001407
1094 004312 000005
1095 004314 000005
1096 004316 004710
1097 004320 000240
1098 004322 000240
1099 004324 000240
1100 004326 000240
1101 004330 000137 002050
1102 004334 000240
1103 004336 000240
1104 004340 000240
1105
1106
1107
1108
1109 004342 104000
1110 004344 012777 004614 174516
1111 004352 013777 001004 174512
1112 004360 012737 000100 001034
1113 004366 012700 012470
1114 004372 012737 000100 001032
1115 004400 012700 117144
1116 004404 012700 000700
1117 004410 012700 000474
1118 004414 004737 004556
1119 004420 012700 173400
1120 004424 012700 160000
1121 004430 012700 012470
1122 004434 005037 005050
1123 004440 012737 030000 012374
1124 004446 012737 030000 012372
1125
1126 004454 005737 005556
1127 004460 001005
1128
1129 004442 004537 005412
1130 004466 000100
1131 004470 011714
1132 004472 000770
1133
1134 004474 004537 005412
1135 004500 000001
1136 004502 012302
1137
1138 004504 004537 005412
1139 004510 000001
1140 004512 012470
1141
1142
1143 004514 005337 001032
1144 004520 001355
1145
1146 004522 005337 001034
1147 004526 001317
1148 004530 000137 004342
1149

```

```

1106
1107
1108
1109 004342 104000
1110 004344 012777 004614 174516
1111 004352 013777 001004 174512
1112 004360 012737 000100 001034
1113 004366 012700 012470
1114 004372 012737 000100 001032
1115 004400 012700 117144
1116 004404 012700 000700
1117 004410 012700 000474
1118 004414 004737 004556
1119 004420 012700 173400
1120 004424 012700 160000
1121 004430 012700 012470
1122 004434 005037 005050
1123 004440 012737 030000 012374
1124 004446 012737 030000 012372
1125
1126 004454 005737 005556
1127 004460 001005
1128
1129 004442 004537 005412
1130 004466 000100
1131 004470 011714
1132 004472 000770
1133
1134 004474 004537 005412
1135 004500 000001
1136 004502 012302
1137
1138 004504 004537 005412
1139 004510 000001
1140 004512 012470
1141
1142
1143 004514 005337 001032
1144 004520 001355
1145
1146 004522 005337 001034
1147 004526 001317
1148 004530 000137 004342
1149

```

```

1151
1152 004534 012701 000030 LOADAC: MOV #24,R1 ILOAD COUNT
1153 004540 012720 130000 MOV #RELATV,(R0)+ ILOAD RELATIVE POINT
1154 004544 012720 040004 15: MOV #INIX*4,(R0)+ ILOAD INTEN BIT
1155 004550 005301 OEC R1 IFINISWEO ?
1156 004552 001374 BNE 15 IBR IF NOT
1157 004554 000207 RTS PC IEXIT?
1158
1159 004556 012737 000030 001046 LOAOU: MOV #24,CNTR ILOAD COUNT
1160 004564 004737 004534 15: JSR PC,LOAOU ILOAD ACROSS
1161 004570 012720 110000 MOV #LOHGV,(R0)+ ILOAD LONG VECTOR
1162 004574 012720 000004 MOV #4,(R0)+ ILOAD VECTOR OVER
1163 004580 012720 020140 MOV #MINUS*140,(R0)+ IAND UP
1164 004604 005337 001046 OEC CNTR
1165 004610 001365 BNE 15 IBR IF NOT DONE
1166 004612 000207 RTS PC IEXIT
1167
1168
1169 004614 017737 174242 004754 RET14: MOV #YPOS,405
1170 004622 042737 176000 004754 BIC #17000,405
1171 004630 017737 174224 004756 MOV #XPOS,415
1172 004636 042737 176000 004756 BIC #17000,415
1173 004644 005737 005556 TST SWITCM ITEST SW
1174 004650 001411 BEO 15 IBR IF LIGHT PEN FOLLOW
1175 004652 000237 005050 INC HITCNT IUPDATE LIGHT PEN HIT COUNT
1176 004656 013703 005050 MOV HITCNT,R3 ILOAD R3
1177 004662 012702 012376 MOV #FRM160,R2 ILOAD ADDRESS
1178 004666 004737 005332 JSR PC,R0CMR ICONVERT OCTAL
1179 004672 000432 BR 205 IBR
1180 004674 013703 004756 15: MOV 415,R3 ILOAD R3
1181 004700 012702 011770 MOV #DLI140*4,R2 ILOAD ADDRESS
1182 004704 004737 005332 JSR PC,R0CMR ILOAD X READOUT
1183 004710 013703 004754 MOV 405,R3 ILOAD R3
1184 004714 012702 012002 MOV #DLI140*4,R2 ILOAD ADDRESS
1185 004720 004737 005332 JSR PC,R0CMR ILOAD Y READOUT
1186 004724 013737 004754 012006 MOV #05,RAY140 ILOAD NEW Y POSITION
1187 004732 013737 004756 012004 MOV 415,RAY144 ILOAD NEW X POSITION
1188 004740
1189 004748 012777 000001 174106 105: MOV #1,00PC ISINGLE STEP THE DISPLAY
1190 004746 022626 CHP (SP),(SP)+
1191 004750 000137 005430 JMP MESSA
1192 004754 000000 405: 0
1193 004756 000000 415: 0
1194

```

```

1196
1197 004760 005001 205: CLR R1
1198 004762 005002 CLR R2
1199 004764 013700 004756 MOV 415,R0 IGET X AXIS
1200 004770 162700 000700 SUB #700,R0 IGET A BASE ADDRESS
1201 004774 000200 ASR R0
1202 004776 000200 ASR R0
1203 005000 001404 BEO 305
1204 005002 062701 000070 215: AOD #70,R1 IUPDATE OFFSET
1205 005006 005300 OEC R0
1206 005010 001374 BNE 215 IBR UNTIL DONE
1207
1208 005012 013700 004754 305: MOV 405,R0 IGET X AXIS
1209 005016 162700 000500 SUB #500,R0 IMAKE BASE ADDRESS
1210 005022 000200 ASR R0
1211 005024 000200 ASR R0 ISHIFT RIGHT
1212 005026 001404 BEO 325
1213 005030 062701 000002 315: AOD #2,R1
1214 005034 005300 OEC R0
1215 005036 001374 BNE 315
1216 005040 040000 012500 325: BIC #INIX,BUFFER*10(R1) ICLEAR THE BIT
1217 005046 000734 BR 105
1218
1219 005050 000000 HITCNT: 0

```

ECHO ROUTINE KEYBOARD TO DISPLAY			
1221			
1222			
1223	005052	104000	012470
1224	005054	012700	173400
1225	005060	012720	013470
1226	005064	022700	
1227	005070	001373	
1228	005072	005037	001052
1229	005076	005037	001032
1230	005102	112737	000060
1231	005110	112737	000060
1232	005116	112737	000060
1233	005124	112737	000060
1234	005132	012737	000060
1235	005140	012737	000060
1236	005146	052777	000100
1237	005154	012737	000700
1238	005162	012700	012470
1239	005164	004537	005412
1240	005172	000001	
1241	005174	012404	
1242	005176	005737	001052
1243	005202	001010	
1244	005204	000770	
1245	005206	017701	173602
1246	005212	042701	177600
1247	005216	012737	177777
1248	005224	000002	
1249	005226	000000	
1250	005230	005037	001052
1251	005234	022701	000003
1252	005240	001002	
1253	005242	000137	001456
1254	005246	005337	005330
1255	005252	001002	
1256	005254	000137	005054
1257	005260	012702	012465
1258	005264	010103	
1259	005266	004737	005332
1260	005272	005737	001032
1261	005276	001007	
1262	005300	110120	
1263	005302	112710	000017
1264	005306	005137	001032
1265	005312	000137	005166
1266	005316	110120	
1267	005320	005037	001032
1268	005324	000137	005166
1269			
1270			
1271	005330	000200	

IUPDATE OCTAL READOUT			
1273			
1274			
1275	005332	042703	176000
1276	005336	004737	005376
1277	005342	110442	
1278	005344	004737	005370
1279	005350	110442	
1280	005352	004737	005370
1281	005356	110442	
1282	005360	004737	005370
1283	005364	110442	
1284	005366	000207	
1285	005370	000003	
1286	005372	000003	
1287	005374	000003	
1288	005376	010304	
1289	005400	042704	177770
1290	005404	062704	000060
1291	005410	000207	
1292			
1293	005412	012537	005552
1294	005416	012537	005554
1295	005422	013777	005554
1296	005430	005077	173354
1297	005434	000001	
1298	005436	005737	002046
1299	005442	001025	
1300	005444	005337	005552
1301	005450	001405	
1302	005452	012777	000001
1303	005460	000137	005430
1304	005464	000240	
1305	005466	005737	002046
1306	005472	001010	
1307	005474	005037	005556
1308	005500	032737	000100
1309	005506	001402	
1310	005510	005137	005556
1311	005514	000205	
1312	005516	005737	005556
1313	005522	001350	
1314	005524	005737	001050
1315	005530	001745	
1316	005532	005037	001050
1317	005536	005037	005556
1318	005542	005037	001042
1319	005546	000137	001162
1320	005552	000000	
1321	005554	000000	
1322	005556	000000	

```

1324
1325 005560 114000 FRME01 POINT
1326 005562 000000 0
1327 005564 001500 MAXV=277
1328 005566 170052 STATA11TALISYNOFFIGREEN
1329 005570 103124 CHAR1INT41LOFF1BLKOFFILINE0
1330 005572 017 017 ,BYTE 17,17
1331 005574 052107 032055 020060 ,ASCII /GT=40 OR GT=44 WITH VR17 VISUAL TEST (MD=11=DDGTG=A)/
005576 051117 043440 026524
005578 032064 053440 052111
005580 020110 051126 033461
005582 053040 051511 040525
005584 020114 042524 052123
005586 020040 046474 026504
005588 030461 042055 043504
005590 043524 040455 000076
1332 005592 015 012 ,BYTE 15,12,12
1333 005594 040 020040 044504 ,ASCII / DIRECTORY/
005596 042522 052103 051117
005598 131
1334 005600 015 012 012 ,BYTE 15,12,12
1335 005602 030060 036440 040440 ,ASCII /00 = A = DIRECTORY/
005604 036440 042040 051111
005606 041505 047524 054522
1336 005608 015 012 ,BYTE 15,12
1337 005610 030460 036440 041040 ,ASCII /01 = B = DOT REPEATIBILITY/
005612 036440 042040 052117
005614 051040 050105 040505
005616 044524 044502 044514
005618 054524
1338 005620 015 012 ,BYTE 15,12
1339 005622 031060 036440 041440 ,ASCII /02 = C = PINCUSHION AND VECTOR CURVATURE (X OR Y OFFSET ADJ.)/
005624 036440 050040 047111
005626 052503 044123 047511
005628 020116 047101 020104
005630 042526 052103 051117
005632 041440 051125 040526
005634 052524 042522 036040
005636 020130 051117 054440
005638 047440 043106 042523
005640 020124 042101 027112
005642 076
1340 005644 015 012 ,BYTE 15,12
1341 005646 060 020063 020075 ,ASCII /03 = D = OCTAGONS OR SQUARES/
005648 020104 020075 041517
005650 040524 047507 051516
005652 047440 020122 050523
005654 040525 042522 123
1342 005656 015 012 ,BYTE 15,12
1343 005658 060 020064 020075 ,ASCII /04 = E = CHARACTER SET (CHAR, ADJ.)/
005660 020105 020075 044103
005662 051101 041501 042524
005664 020122 042523 020124
005666 041474 040510 027122

```

```

1344 005668 040440 045104 037056 ,BYTE 15,12
1345 005670 015 012 ,ASCII /05 = F = DASH LINES AND BLINK/
005672 032460 036440 043040
005674 036440 042040 051501
005676 020110 044514 042516
005678 020123 047101 020104
005680 046102 047111 113
1346 005682 015 012 ,BYTE 15,12
1347 005684 060 020066 020075 ,ASCII /06 = G = HORIZONTAL VECTOR ANGLE (ADJ. X VECTOR LENGTH)/
005686 020107 020075 047510
005688 044522 047532 052116
005690 046101 053040 041505
005692 047524 020122 047101
005694 046107 020105 040474
005696 045104 020056 020130
005698 042526 052103 051117
005700 046040 047105 052107
005702 037110
1348 005704 015 012 ,BYTE 15,12
1349 005706 033460 036440 044040 ,ASCII /07 = H = VERTICAL VECTOR ANGLE (ADJ. Y VECTOR LENGTH)/
005708 036440 053040 051105
005710 044524 040503 020114
005712 042526 052103 051117
005714 040440 043516 042514
005716 036040 042101 027112
005718 054440 053040 041505
005720 047524 020122 042514
005722 043516 044124 076
1350 005724 015 012 ,BYTE 15,12
1351 005726 061 020060 020075 ,ASCII /10 = I = HORIZONTAL PHOSPHOR TEST/
005728 020111 020075 047510
005730 044522 047532 052116
005732 046101 050040 047510
005734 050123 047510 020122
005736 042524 052123
1352 005738 015 012 ,BYTE 15,12
1353 005740 030461 036440 045040 ,ASCII /11 = J = VERTICAL PHOSPHOR TEST/
005742 036440 053040 051105
005744 044524 040503 020114
005746 044120 051517 044120
005748 051117 052040 051505
005750 124
1354 005752 015 012 ,BYTE 15,12
1355 005754 061 020062 020075 ,ASCII /12 = K = INTENSITY LEVEL AND LIGHT-PEN TEST/
005756 020113 020075 047111
005758 042524 051516 052111
005760 020131 042514 042526
005762 020114 047101 020104
005764 044514 044107 026524
005766 042520 020116 042524
005768 052123
1356 005770 015 012 ,BYTE 15,12
1357 005772 031461 036440 046040 ,ASCII /13 = L = EDGE FLAG TEST/
005774 036440 042440 043504

```

1358	006602	020105	046106	043501	
	006610	052040	051505	124	
1359	006615	015	012		.BYTE 15,12
	006617	061	020064	020075	.ASCII /14 = M = SHORT VECTORS AND RELATIVE POINT,
	006624	020117	020075	044123	
	006632	051117	020124	042506	
	006640	052103	051117	020123	
	006646	047101	020104	042522	
	006654	040514	044524	042526	
	006662	050040	044517	052116	
1360	006670	015	012		.BYTE 15,12
1361	006672	032461	036440	047040	.ASCII /15 = N = GRAPHLOT TEST/
	006670	036440	043440	040522	
	006676	044120	046120	052117	
	0066714	052040	051505	124	
1362	0066721	015	012		.BYTE 15,12
1363	0066723	061	020066	020075	.ASCII /16 = D = LIGHT PEN FOLLOW,
	0066730	020117	020075	044514	
	0066736	044107	020124	042520	
	0066744	020116	047506	046114	
	0066752	053517			
1364	0066754	015	012		.BYTE 15,12
1365	0066776	033461	036440	050040	.ASCII /17 = P = KEYBOARD ECHO TEST/
	0066764	036440	045440	054505	
	0066772	047502	051101	020104	
	0067000	041505	047510	052040	
	0067006	051505	124		
1366	0067011	015	012		.BYTE 15,12,12
1367	0067014	020040	052522	047502	.ASCII / KUBDUY TO REMAIN ON THE PATTERN,
	0067022	052125	052040	020117	
	0067030	042522	040515	047111	
	0067036	047440	020116	044124	
	0067044	020105	040520	052124	
	0067052	051105	116		
1368	0067055	015	012		.BYTE 15,12
1369	0067057	040	041440	020122	.ASCII / CR TO SELECT SUB-PICTURE OR STOP MOTION /
	0067064	047524	051440	046105	
	0067072	041505	020124	052523	
	0067100	026502	044520	052103	
	0067106	051125	020105	051117	
	0067114	051440	047524	020120	
	0067122	047515	044524	047117	
	0067130	020040			
1370					.EVEN
1371	0067132	173400			DSTOP
1372	0067134	160000			DJMP
1373	0067136	005500			FRME0
1374					
1375	0067140				FRME1
(1)	0067140	170052			STATSAIITALEISYNDOFFIGREEN
1376	0067142	116124			POINTIINT01LPOFFIBLKOFFLINE0
1377	0067144	041000			INTX=1000
1378	0067146	001000			MAXY=1/2
1379	0067150	040000			INTX=0

1380	0067152	000000			0
1381	0067154	041000			INTX=1000
1382	0067156	001000			MAXY=1/2
1383	0067160	041777			INTX=1777
1384	0067162	000000			0
1385	0067164	041000			INTX=1000
1386	0067166	001000			MAXY=1/2
1387	0067170	041777			INTX=1777
1388	0067172	001777			MAXY
1389	0067174	041000			INTX=1000
1390	0067176	001000			MAXY=1/2
1391	0067200	040000			INTX
1392	0067202	001777			MAXY
1393	0067204	164000			DNOP
1394	0067206	164000			DNOP
1395	0067210	164000			DNOP
1396	0067212	164000			DNOP
1397	0067214	164000			DNOP
1398	0067216	164000			DNOP
1399	0067220	164000			DNOP
1400	0067222	173400			DSTOP
1401	0067224	160000			DJMP
1402	0067226	0067140			FRME1
1403					
1404					
1405					
1406	0067230	116524			FRME2
1407	0067232	000000			POINTIINT2LPOFFIBLKOFFLINE0
1408	0067234	000000			0
1409	0067236	170052			STATSAIITALEISYNDOFFIGREEN
1410	0067240	110000			LONGV
1411	0067242	041777			INTXIMAXX
1412	0067244	000000			0
1413	0067246	040000			INTX
1414	0067250	001777			MAXY
1415	0067252	061777			INTXIMINUSXIMAXX
1416	0067254	000000			0
1417	0067256	040000			INTX
1418	0067260	021777			MINUSYIMAXY
1419	0067262	041777			INTXIMAXX
1420	0067264	020000			MINUSY
1421	0067266	060000			INTXIMINUSX
1422	0067270	001777			MAXY
1423	0067272	061777			INTXIMINUSXIMAXX
1424	0067274	020000			MINUSY
1425	0067276	060000			INTXIMINUSX
1426	0067300	021777			MINUSYIMAXY
1427	0067302	041777			INTXIMAXX
1428	0067304	001777			MAXY
1429	0067306	061777			INTXIMINUSXIMAXX
1430	0067310	021777			MINUSYIMAXY
1431	0067312	001777			MAXX
1432	0067314	000000			0
1433	0067316	061777			INTXIMINUSXIMAXX

1434 027320 001777
 1435 027322 241777
 1436 027324 021777
 1437 027326 173402
 1438 027330 160200
 1439 027332 007230
 1440
 1441
 1442
 1443 027334 117124
 1444 027336 000774
 1445 027340 000564
 1446 027342 170052
 1466 027344 110000
 (1) 027346 040007
 (1) 027350 000000
 (1) 027352 040007
 (1) 027354 000007
 (1) 027356 040000
 (1) 027360 000007
 (1) 027362 040007
 (1) 027364 000007
 (1) 027366 040007
 (1) 027370 000000
 (1) 027372 040007
 (1) 027374 020007
 (1) 027376 040000
 (1) 027400 020007
 (1) 027402 040007
 (1) 027404 020007
 1467 027406 114000
 1468 027410 000770
 1469 027412 000550
 1470 027414 110000
 (1) 027416 040017
 (1) 027420 000000
 (1) 027422 040017
 (1) 027424 000017
 (1) 027426 040000
 (1) 027430 000017
 (1) 027432 040017
 (1) 027434 000017
 (1) 027436 040017
 (1) 027440 000000
 (1) 027442 040017
 (1) 027444 020017
 (1) 027446 040000
 (1) 027450 020017
 (1) 027452 040017
 (1) 027454 020017
 1471 027456 114000
 1472 027460 000762
 1473 027462 000520
 1474 027464 110000

MAXY
 INTXIMAXX
 MINUSXIMAXY
 OSTOP
 OJMP
 FRME2
 OCTAGONS
 FRME3, POINT,INT4,LP OFF,BLK OFF,LINE0
 774
 564
 STAYSAIITALEISYN OFFIGREEN
 LONGV
 INTX*7
 0
 INTX*7
 7
 INTX
 7
 INTXIMINUSX*7
 7
 INTXIMINUSX*7
 0
 INTXIMINUSX*7
 MINUSX*7
 INTX
 MINUSX*7
 INTX*7
 MINUSX*7
 POINT
 770
 550
 LONGV
 INTX*17
 0
 INTX*17
 17
 INTX
 17
 INTXIMINUSX*17
 17
 INTXIMINUSX*17
 0
 INTXIMINUSX*17
 MINUSX*17
 INTX
 MINUSX*17
 INTX*17
 MINUSX*17
 POINT
 762
 520
 LONGV
 OCTOGON BY LENGTH OF 7
 OCTOGON BY LENGTH OF 17
 OCTOGON BY LENGTH OF 37

(1) 027466 040037
 (1) 027470 000000
 (1) 027472 040037
 (1) 027474 000037
 (1) 027476 040000
 (1) 027500 000037
 (1) 027502 040037
 (1) 027504 000037
 (1) 027506 040037
 (1) 027510 000000
 (1) 027512 040037
 (1) 027514 020037
 (1) 027516 040000
 (1) 027520 020037
 (1) 027522 040037
 (1) 027524 020037
 1475 027526 114000
 1476 027530 000740
 1477 027532 000440
 1478 027534 110000
 (1) 027536 040077
 (1) 027540 000000
 (1) 027542 040077
 (1) 027544 000077
 (1) 027546 040000
 (1) 027550 000077
 (1) 027552 040077
 (1) 027554 000077
 (1) 027556 040077
 (1) 027560 000000
 (1) 027562 040077
 (1) 027564 020077
 (1) 027566 040000
 (1) 027570 020077
 (1) 027572 040077
 (1) 027574 020077
 1479 027576 114000
 1480 027600 000700
 1481 027602 000300
 1482 027604 110000
 (1) 027606 040177
 (1) 027610 000000
 (1) 027612 040177
 (1) 027614 000177
 (1) 027616 040000
 (1) 027620 000177
 (1) 027622 040177
 (1) 027624 000177
 (1) 027626 040177
 (1) 027630 000000
 (1) 027632 040177
 (1) 027634 020177
 (1) 027636 040000
 (1) 027640 020177

INTX*37
 0
 INTX*37
 37
 INTX
 37
 INTXIMINUSX*37
 37
 INTXIMINUSX*37
 0
 INTXIMINUSX*37
 MINUSX*37
 INTX
 MINUSX*37
 INTX*37
 MINUSX*37
 POINT
 740
 440
 LONGV
 INTX*77
 0
 INTX*77
 77
 INTX
 77
 INTXIMINUSX*77
 77
 INTXIMINUSX*77
 0
 INTXIMINUSX*77
 MINUSX*77
 INTX
 MINUSX*77
 INTX*77
 MINUSX*77
 POINT
 700
 300
 LONGV
 INTX*177
 0
 INTX*177
 177
 INTX
 177
 INTXIMINUSX*177
 177
 INTXIMINUSX*177
 0
 INTXIMINUSX*177
 MINUSX*177
 INTX
 MINUSX*177
 OCTOGON BY LENGTH OF 77
 OCTOGON BY LENGTH OF 177

(1)	007642	040177	INTX*177	
(1)	007644	020177	MINUSX*177	
1483	007646	114000	POINT	
1484	007650	000600	000	
1485	007652	000000	0	
1486	007654	110000	LONGV	10CTDGN BY LENGTH OF 377
(1)	007656	040377	INTX*377	
(1)	007660	000000	0	
(1)	007662	040377	INTX*377	
(1)	007664	000377	377	
(1)	007666	040000	INTX	
(1)	007670	000377	377	
(1)	007672	060377	INTX:MINUSX*377	
(1)	007674	000377	377	
(1)	007676	060377	INTX:MINUSX*377	
(1)	007700	000000	0	
(1)	007702	060377	INTX:MINUSX*377	
(1)	007704	020377	MINUSX*377	
(1)	007706	040000	INTX	
(1)	007710	020377	MINUSX*377	
(1)	007712	040377	INTX*377	
(1)	007714	020377	MINUSX*377	
1487	007716	173400	OSTOP	
1488	007720	160000	0JMP	
1489	007722	007334	FRME3	
1490			ISQUARES	7,17,37,77,177,327,777 WIDE
1491				
1492	007724	117124	FRME3A9	POINT:INT41,POFF:BLKOFF:LINE0
1493	007726	001000		BY 7
1494	007730	000600	000	
1495	007732	170032	STATSA:ITAL:ISY:OFF:GREEN	
1509		000007	Q=7	
1510		000004	R=4	
1517	007734	110000	LONGV	BY 7 AND 4
(2)	007736	040007	INTX*7	
(2)	007740	000000	0	
(2)	007742	040000	INTX	
(2)	007744	000007	7	
(2)	007746	060007	INTX:MINUSX*7	
(2)	007750	000000	0	
(2)	007752	040000	INTX	
(2)	007754	020007	MINUSX*7	
(2)	007756	020004	MINUSX*4	
(2)	007760	020004	MINUSX*4	
(1)			LIST	
(2)	007762	110000	LONGV	BY 17 AND 7
(2)	007764	040017	INTX*17	
(2)	007766	000000	0	
(2)	007770	040000	INTX	
(2)	007772	000017	17	
(2)	007774	060017	INTX:MINUSX*17	
(2)	007776	000000	0	
(2)	010000	040000	INTX	
(2)	010002	020017	MINUSX*17	

(2)	010004	020007	MINUSX*7	
(2)	010006	020007	MINUSX*7	
(1)			LIST	
(2)	010010	110000	LONGV	BY 37 AND 17
(2)	010012	040037	INTX*37	
(2)	010014	000000	0	
(2)	010016	040000	INTX	
(2)	010020	000037	37	
(2)	010022	060037	INTX:MINUSX*37	
(2)	010024	000000	0	
(2)	010026	040000	INTX	
(2)	010030	020037	MINUSX*37	
(2)	010032	020017	MINUSX*17	
(2)	010034	020017	MINUSX*17	
(1)			LIST	
(2)	010036	110000	LONGV	BY 77 AND 37
(2)	010040	040077	INTX*77	
(2)	010042	000000	0	
(2)	010044	040000	INTX	
(2)	010046	000077	77	
(2)	010050	060077	INTX:MINUSX*77	
(2)	010052	000000	0	
(2)	010054	040000	INTX	
(2)	010056	020077	MINUSX*77	
(2)	010060	020037	MINUSX*37	
(2)	010062	020037	MINUSX*37	
(1)			LIST	
(2)	010064	110000	LONGV	BY 177 AND 77
(2)	010066	040177	INTX*177	
(2)	010070	000000	0	
(2)	010072	040000	INTX	
(2)	010074	000177	177	
(2)	010076	060177	INTX:MINUSX*177	
(2)	010080	000000	0	
(2)	010082	040000	INTX	
(2)	010084	020177	MINUSX*177	
(2)	010086	020077	MINUSX*77	
(2)	010088	020077	MINUSX*77	
(1)			LIST	
(2)	010092	110000	LONGV	BY 377 AND 177
(2)	010094	040377	INTX*377	
(2)	010096	000000	0	
(2)	010098	040000	INTX	
(2)	010102	000377	377	
(2)	010104	060377	INTX:MINUSX*377	
(2)	010106	000000	0	
(2)	010108	040000	INTX	
(2)	010112	020377	MINUSX*377	
(2)	010114	020177	MINUSX*177	
(2)	010116	020177	MINUSX*177	
(1)			LIST	
(2)	010120	110000	LONGV	BY 777 AND 377
(2)	010122	040777	INTX*777	
(2)	010124	000000	0	
(2)	010126	040000	INTX	
(2)	010130	040000	INTX	
(2)	010132	020377	MINUSX*377	
(2)	010134	020177	MINUSX*177	
(2)	010136	020177	MINUSX*177	
(1)			LIST	
(2)	010140	110000	LONGV	BY 777 AND 377
(2)	010142	040777	INTX*777	
(2)	010144	000000	0	

```

(2) 011146 040000 INTX
(2) 011150 000777 777
(2) 011152 000777 INTXMINUSX*777
(2) 011154 000000 0
(2) 011156 040000 INTX
(2) 011160 020777 MINUSX*777
(2) 011162 020377 MINUSX*377
(2) 011164 020377 MINUSX*377
(1) 011166 173400 ,LIST
1518 011170 160000 OSTOP
1519 011172 007724 OJMP
1520 011172 007724 FRMESA
1521
1522 IDASH LINE TEST
1523
1524 011174 117000 FRMESA POINTINT4
1525 011176 000000 0
1526 011200 001000 1000
1527 011202 174400 STATA:ITAL%ISYNOFFIGREEN
1528 011204 170052 STATA:ITAL%ISYNOFFIGREEN
1529 011206 100004 CHARILINE0
1530 011210 017 ,BYTE 17,17
1531 011212 047523 044514 020104 ,ASCII /SOLJO /
011220 020040 020040
1532 011224 110004 LONGVILINE0
1533 011226 040400 40400
1534 011230 000000 0
1535 011232 000400 400
1536 011234 000000 0
1537 011236 110030 LONGVIBLKON
1538 011240 040400 40400
1539 011242 000000 0
1540 011244 100020 CHARIBLKOFF
1541 011246 015 012 012 ,BYTE 15,12,12,12,12,12
011251 012 012 012
1542 011254 040504 044123 044440 ,ASCII /DASH I /
011262 020040 020040
1543 011266 110005 LONGVILINE1
1544 011270 040400 40400
1545 011272 000000 0
1546 011274 000400 400
1547 011276 000000 0
1548 011300 110030 LONGVIBLKON
1549 011302 040400 40400
1550 011304 000000 0
1551 011306 100020 CHARIBLKOFF
1552 011308 015 012 012 ,BYTE 15,12,12,12,12,12
011313 012 012 012
1553 011316 040504 044123 044440 ,ASCII /DASH II /
011324 020111 020040
1554 011326 110006 LONGVILINE2
1555 011330 040400 40400
1556 011332 000000 0
1557 011336 000400 400

```

```

1558 011340 000000 0
1559 011342 110030 LONGVIBLKON
1560 011344 040400 40400
1561 011346 000000 0
1562 011350 100020 CHARIBLKOFF
1563 011352 015 012 012 ,BYTE 15,12,12,12,12,12
011355 012 012 012
1564 011360 040504 044123 044440 ,ASCII /DASH III /
011366 044511 020040
1565 011372 110007 LONGVILINE3
1566 011374 040400 40400
1567 011376 000000 0
1568 011400 000400 400
1569 011402 000000 0
1570 011404 110030 LONGVIBLKON
1571 011406 040400 40400
1572 011410 000000 0
1573 011412 110024 LONGVIBLKOFFILINE0
1574 011414 000000 0
1575 011416 000000 0
1576 011420 173400 OSTOP
1577 011422 160000 OJMP
1578 011424 010174 FRMESA
1579
1580 VECTOR LENGTH TEST <FILE 6 AND 7>
1581
1582 011426 114000 FRMESA POINT
1583 011430 001777 MAXX
1584 011432 000000 0
1585 011434 170052 STATA:ITAL%ISYNOFFIGREEN
1586 011436 113724 LONGV:INT7ILPOFFIBLKOFFILINE0
1587 011440 040000 INTX
1588 011442 001777 MAXY
1589 011444 114000 POINT
1590 011446 000000 0
1591 011450 001777 MAXY
1592 011452 110000 LONGV
1593 011454 041777 INTX:MAXX
1594 011456 000000 0
1595 011460 173400 OSTOP
1596 011462 114000 FRMESA POINT
1597 011464 000000 0
1598 011466 000000 0
1599 011470 110000 LONGV
1600 011472 000000 DELTX6: 0
1601 011474 000000 DELTY6: 0
1602 011476 173400 OSTOP
1603 011500 160000 OJMP
1604 011502 010462 FRMESA
1605
1606 IPMOSPHOR TEST
1607
1608
1609 011504 114000 FRMESA POINT

```

```

1610 010506 000000 DELTX7: 0
1611 010510 000000 0
1612 010512 170002 STATA:ITALB:SYNOFF:GREEN
1613 010514 113724 DF110A: LONGV:INT7:LBKOFF:LINE0
1614 010516 040000 INTX
1615 010520 001777 MAXY
1616 010522 000002 2
1617 010524 000000 0
1618 010526 040000 INTX
1619 010530 021777 MINUSY:MAXY
1620 010532 000002 2
1621 010534 000000 0
1622 010536 173400 OSTOP
1623 010540 160000 OJMP
1624 010542 010514 DF110A
1625
1626
1627
1628 010544 114000 IPHOSPHOR TEST
1629 010546 000000 FRME11: POINT
1630 010550 000000 0
1631 010552 170052 DELTY7: 0
1632 010554 113724 STATA:ITALB:SYNOFF:GREEN
1633 010556 041777 DF111C: LONGV:INT7:LBKOFF:LINE0
1634 010560 000000 INTX:MAXX
1635 010562 000000 0
1636 010564 000002 2
1637 010566 061777 INTX:MINUSX:MAXX
1638 010570 000000 0
1639 010572 000000 0
1640 010574 000002 2
1641 010576 173400 OSTOP
1642 010600 160000 OJMP
1643 010602 010554 DF111C
1644
1645 010604 117604 FRM10: POINT:INT7:LINE0
1646 010606 000000 0
1647 010610 000000 0
1648 010612 110000 LONGV
1649 010614 041777 INTX:MAXX
1650 010616 000000 0
1651 010620 040000 INTX
1652 010622 001777 MAXY
1653 010624 061777 INTX:MINUSX:MAXX
1654 010626 000000 0
1655 010630 040000 INTX
1656 010632 021777 MINUSX:MAXY
1657 010634 173400 OSTOP
1658 010636 160000 OJMP
1659 010640 010604 FRM10
1660
1661
1662
1663 010642 114164 J:INTENSITY TEST
FRME12: POINT:LINEB:LPON:BLKOFF

```

```

1664 010644 000000 0
1665 010646 001200 1200
1666 010650 170252 SYN12: STATA:ILPLI:ISYNOFF:ITALB:GREEN
1667 010652 103600 CHAR:INT7
1668 010654 017 .BYTE 17,17
1669 010656 047111 042524 051516 .ASCII /INTENSITY 7 /
010664 052111 020131 020067
010672 020040
1670 010674 110000 LONGV
1671 010676 041000 41000
1672 010700 000000 0
1673 010702 130000 RELATV
1674 010704 057600 57600
1675 010706 103400 CHAR:INT6
1676 010710 015 012 012 .BYTE 15,12,12,12
010713 012
1677 010714 047111 042524 051516 .ASCII /INTENSITY 6 /
010722 052111 020131 020066
010730 020040
1678 010732 110000 LONGV
1679 010734 041000 41000
1680 010736 000000 0
1681 010740 130000 RELATV
1682 010742 057600 57600
1683 010744 103200 CHAR:INT5
1684 010746 015 012 012 .BYTE 15,12,12,12
010751 012
1685 010752 047111 042524 051516 .ASCII /INTENSITY 5 /
010760 052111 020131 020065
010766 020040
1686 010770 110000 LONGV
1687 010772 041000 41000
1688 010774 000000 0
1689 010776 130000 RELATV
1690 011000 057600 57600
1691 011002 103000 CHAR:INT4
1692 011004 015 012 012 .BYTE 15,12,12,12
011007 012
1693 011010 047111 042524 051516 .ASCII /INTENSITY 4 /
011016 052111 020131 020064
011024 020040
1694 011026 110000 LONGV
1695 011030 041000 41000
1696 011032 000000 0
1697 011034 130000 RELATV
1698 011036 057600 57600
1699 011040 102600 CHAR:INT3
1700 011042 015 012 012 .BYTE 15,12,12,12
011045 012
1701 011046 047111 042524 051516 .ASCII /INTENSITY 3 /
011054 052111 020131 020063
011062 020040
1702 011064 110000 LONGV
1703 011066 041000 41000

```

1704 011070 000000 0
 1705 011072 130000 RELATV
 1706 011074 057600 57600
 1707 011076 102400 CHARIN#2
 1708 011100 015 012 012 ,BYTE 15,12,12,12
 1709 011103 012 012
 011104 047111 042524 051516 ,ASCII /INTENSITY 2 /
 011112 052111 020131 020062
 011120 020040
 1710 011122 110000 LONGV
 1711 011124 041000 41000
 1712 011126 000000 0
 1713 011130 130000 RELATV
 1714 011132 057600 57600
 1715 011134 102200 CHARIN#1
 1716 011136 015 012 012 ,BYTE 15,12,12,12
 011141 012
 1717 011142 047111 042524 051516 ,ASCII /INTENSITY 1 /
 011150 052111 020131 020061
 011156 020040
 1718 011160 110000 LONGV
 1719 011162 041000 41000
 1720 011164 000000 0
 1721 011166 130000 RELATV
 1722 011170 057600 57600
 1723 011172 102000 CHARIN#0
 1724 011174 015 012 012 ,BYTE 15,12,12,12
 011177 012
 1725 011200 047111 042524 051516 ,ASCII /INTENSITY 0 /
 011206 052111 020131 020060
 011214 020040
 1726 011216 110000 LONGV
 1727 011220 041000 41000
 1728 011202 000000 0
 1729 011204 130000 RELATV
 1730 011206 057600 57600
 1731 011230 164000 ONOP
 1732 011232 164000 ONOP
 1733 011234 164000 ONOP
 1734 011236 164000 ONOP
 1735 011240 164000 ONOP
 1736 011242 164000 ONOP
 1737 011244 164000 ONOP
 1738 011246 164000 ONOP
 1739 011250 173400 RAYLPA# DSTOP
 1740 011252 164000 ONOP
 1741 011254 164000 ONOP
 1742
 1743 011256 117100 OFI12A# POINT:INT4:LP OFF
 1744 011260 001500 1500
 1745 011202 001200 LPPNT: 1200
 1746 011204 100000 CHAR
 1747 011206 044514 044107 020124 ,ASCII /LIGHT PEN HIT/
 011274 042520 020116 044510

1748 011302 000124 ,EVEN
 1749 011304 173400 DSTOP
 1750 011306 160000 OJMP
 1751 011310 010642 FRAME12
 1752
 1753 ,EOGE FILE
 1754
 1755 011312 117124 FRAME13# POINT:INT4:LP OFF:BLK OFF:LINE0
 1756 011314 000000 0
 1757 011316 000000 0
 1758 011320 170052 STATA:ITAL#SYN OFF:IGREEN
 1759 011322 100000 CHAR
 1760 011324 017 017 ,BYTE 17,17
 1761 011326 110000 LONGV
 1762 011330 041777 INTX:MAXX
 1763 011332 000000 0
 1764 011334 040020 INTX
 1765 011336 001777 MAXY
 1766 011340 061777 INTX:MINUSX:MAXX
 1767 011342 000000 0
 1768 011344 040000 INTX
 1769 011346 021777 MINUSY:MAXY
 1770 011350 114000 POINT
 1771 011352 000100 100
 1772 011354 000300 300 ,LEFT SIDE
 1773 011356 110000 LONGV
 1774 011360 040000 INTX
 1775 011362 000400 400
 1776 011364 060200 INTX:MINUSX*200
 1777 011366 000000 0
 1778 011370 040000 INTX
 1779 011372 020400 MINUSY*400
 1780 011374 040200 INTX*200
 1781 011376 000000 0
 1782 011400 114000 POINT
 1783 011402 000200 200 ,TOP SIDE
 1784 011404 001700 MAXY*1=100
 1785 011406 110000 LONGV
 1786 011410 040400 INTX*400
 1787 011412 000000 0
 1788 011414 040000 INTX
 1789 011416 000200 200
 1790 011420 060400 INTX:MINUSX*400
 1791 011422 000000 0
 1792 011424 040000 INTX
 1793 011426 020200 MINUSY*200
 1794 011430 114000 POINT
 1795 011432 001700 1700 ,RIGHT SIDE
 1796 011434 001500 MAXY*1=300
 1797 011436 110000 LONGV
 1798 011440 040000 INTX
 1799 011442 020400 MINUSY*400
 1800 011444 040200 INTX*200

1801	011446	000000	0
1802	011430	040000	INTX
1803	011432	000400	400
1804	011434	060200	INTX:MINUSX*200
1805	011436	000000	0
1806	011460	110000	POINT
1807	011462	001600	1600
1808	011464	000100	100
1809	011466	110000	LONGV
1810	011470	060400	INTX:MINUSX*400
1811	011472	000000	0
1812	011474	040000	INTX
1813	011476	020200	MINUSY*200
1814	011500	040400	INTX*400
1815	011502	000000	0
1816	011504	040000	INTX
1817	011506	000200	200
1818	011510	110000	POINT
1819	011512	001777	MAXX
1820	011514	000400	400
1821	011516	110000	LONGV
1822	011520	000020	20
1823	011522	000000	0
1824	011524	100000	CHAR
1825	011526	015	101
1826	011530	110000	15,101
1827	011532	000000	POINT
1828	011534	000500	500
1829	011536	110000	LONGV
1830	011540	020012	MINUSX*12
1831	011542	000000	0
1832	011544	100000	CHAR
1833	011546	040	102
1834	011550	160000	16,102
1835	011552	160000	DNOP
1836	011554	173400	DNOP
1837	011556	160000	DSTOP
1838	011560	160000	DNOP
1839	011562	160000	DNOP
1840	011564	011312	DJMP
1841			FRME13

180TTDM SIDE

1"CR" AND AN "A"

1"SPACE" AND AN "B"

1843			
1857			
1858	011566	170052	FRME14: STATSAITALBISYNDFJGREEN
1859	011570	117124	POINT:INT4IBLKOFFILPCFFILIN0
1860	011572	000000	FRM14: 0
1861	011574	000000	FRM14B: 0
1862	011576	100000	SHORTV
1863	011600	056200	INTX*16200
(1)	011602	056271	INTX*16200*71
(1)	011604	040071	INTX*71
(1)	011606	076271	INTX:MINUSX*16200*71
(1)	011610	076200	INTX:MINUSX*16200
(1)	011612	076371	INTX:MINUSX*16200*MINUSY*71
(1)	011614	040171	INTX*MINUSY*71
(1)	011616	056371	INTX*16200*MINUSY*71
(1)	011620	020504	20504
(1)	011622	160000	DNOP
(1)	011624	160000	DNOP
1864	011626	130000	RELATV
1865	011630	057000	INTX*17000
(1)	011632	057074	INTX*17000*74
(1)	011634	040074	INTX*74
(1)	011636	077074	INTX:MINUSX*17000*74
(1)	011640	077000	INTX:MINUSX*17000
(1)	011642	077174	INTX:MINUSX*17000*MINUSY*74
(1)	011644	040174	INTX*MINUSY*74
(1)	011646	057174	INTX*17000*MINUSY*74
(1)	011650	020504	20504
(1)	011652	160000	DNOP
(1)	011654	160000	DNOP
1866	011656	100000	SHORTV
1867	011660	057600	INTX*17600
(1)	011662	057677	INTX*17600*77
(1)	011664	040077	INTX*77
(1)	011666	077677	INTX:MINUSX*17600*77
(1)	011670	077600	INTX:MINUSX*17600
(1)	011672	077777	INTX:MINUSX*17600*MINUSY*77
(1)	011674	040177	INTX*MINUSY*77
(1)	011676	057777	INTX*17600*MINUSY*77
(1)	011700	020504	20504
(1)	011702	160000	DNOP
(1)	011704	160000	DNOP
1868	011706	173400	DSTOP
1869	011710	160000	DJMP
1870	011712	011566	FRME14

[illegible]

```

1902 011114 164000 DNOP
1903 011116 164000 DNOP
1904 011120 164000 DNOP
1905 011122 164000 DNOP
1906 011124 130000 RELATV
1907 011126 034000 MINUSX*14000
1908 011130 040160 INTXIMINSUY*60
1911 011132 040004 INTX*4
(1) 011134 040004 INTX*4
(1) 011136 040004 INTX*4
(1) 011140 040004 INTX*4
(1) 011142 040004 INTX*4
(1) 011144 040004 INTX*4
(1) 011146 040004 INTX*4
(1) 011150 040004 INTX*4
(1) 011152 040004 INTX*4
(1) 011154 040004 INTX*4
(1) 011156 040004 INTX*4
(1) 011160 040004 INTX*4
(1) 011162 040004 INTX*4
(1) 011164 040004 INTX*4
(1) 011166 040004 INTX*4
(1) 011170 040004 INTX*4
(1) 011172 040004 INTX*4
(1) 011174 040004 INTX*4
(1) 011176 040004 INTX*4
(1) 011200 040004 INTX*4
(1) 011202 040004 INTX*4
(1) 011204 040004 INTX*4
(1) 011206 040004 INTX*4
(1) 011210 040004 INTX*4
1912 011212 164000 DNOP
1913 011214 164000 DNOP
1914 011216 164000 DNOP
1915 011220 164000 DNOP
1916 011222 164000 DNOP
1917 011224 164000 DNOP
1918 011226 164000 DNOP
1919 011230 164000 DNOP
1920 011232 110000 LDNGV
1921 011234 000000 7
1922 011236 000077 2
1923 011240 040160 PNT16A1 INTX*160
1924 011242 020160 MINUSX*160
1925 011244 060160 PNT16B1 INTXIMINUSX*160
1926 011246 020160 MINUSX*160
1927 011250 060160 PNT16C1 INTXIMINUSX*160
1928 011252 000160 160
1929 011254 040160 PNT16D1 INTX*160
1930 011256 000160 160
1931 011260 164000 DNOP
1932 011262 164000 DNOP
1933 011264 164000 DNOP
1934 011266 164000 DNOP

```

```

1935 012270 164000 ONOP
1936 012272 164000 ONOP
1937 012274 173400 OSTOP
1938 012276 160000 OJMP
1939 012300 011714 FRME16
1940
1941 012302 117124 FRM16A POINT,INT4,LOFF,BLKOFF,LINE0
1942 012304 000000 0
1943 012306 001600 MAXY=177
1944 012310 170000 STATSAI,ITAL,ISYNOFF,IGREEN
1945 012312 100000 CHAR
1946 012314 017 017 .BYTE 17,17
1947 012316 044514 044107 020124 .ASCII /LIGHT PEN FIELD OF VIEW /
012324 042520 020116 044506
012302 046105 020104 043117
012340 053040 042511 020127
1948 012346 015 012 012 .BYTE 15,12,12
1949 012351 116 046525 042502 .ASCII /NUMBER OF HITS = 0000/
012336 020122 043117 044040
012304 052111 020123 020075
012372 030000 030000
1950 012376 173400 FRM16B OSTOP
1951 012400 160000 OJMP
1952 012402 012302 FRM16A
1953
1954 012404 114124 FRM17 POINT,LOFF,BLKOFF,LINE0
1955 012406 000000 0
1956 012410 001600 MAXY=177
1957 012412 170000 STATSAI,ITAL,ISYNOFF,IGREEN
1958 012414 103000 CHAR,INT4
1959 012416 017 017 .BYTE 17,17
1960 012420 042513 041131 040517 .ASCII /KEYBOARD ECHO TEST/
012406 042122 042440 044103
012404 020117 042524 052123
012442 000
1961 012443 015 012 012 .BYTE 15,12,12
1962 012446 044103 051101 047440 .ASCII /CHAR OCT = /
012454 052103 036440 040
1963 012401 000 000 .BYTE 0,0,0,0
012404 000
1964 012405 015 012 012 KBOCT, .BYTE 15,12,12
1965
1966 012470 164000 BUFFER, ONOP
1967
1968 000001 .END

```

MLST BE JUST BEFORE THE BUFFER

```

BIT6 = 000000 BLKOFF = 000020 BLKON = 000030 BUFFER = 012570
CHANGE 001050 CHAR = 100000 CHRCNT = 000330 CNTR = 001040
COUNT 005052 CRLF = 002070 DBUF = 001010 DBUF1 = 001020
DBUF2 001020 DBUF3 = 001024 DBUF4 = 001020 DBUF5 = 001030
DOONE 001064 DOONE1 = 001066 DELTX0 = 010472 DELTX7 = 010500
DBLTY6 010474 DELTY7 = 010550 OFI10A = 010514 OFI11C = 010550
DPI12A = 011256 OFI15B = 004242 OFI15C = 004144 OFI15D = 004220
DISPLA = 117570 OIBPTC = 001210 OJMP = 160000 OLT14A = 011760
DLTY4B 011770 ONOP = 164000 OPC = 001054 OSAVE = 001032
OSAVE1 001034 OSAVE2 = 001036 OSAVE3 = 001040 OSR = 001054
OSTOP = 173400 OYA = 003330 OYC = 003360 OYD = 003400
OYF = 003434 ECHOA = 005060 ECHOB = 005310 ECHOC = 005160
FILE = 005054 FILE0 = 002050 FILE1 = 002062 FILE10 = 003322
FILE11 = 003378 FILE12 = 003452 FILE13 = 003614 FILE14 = 003624
FILE15 = 004104 FILE16 = 004342 FILE17 = 005052 FILE2 = 002074
FILE2A = 002274 FILE2B = 002302 FILE2C = 003322 FILE2D = 002330
FILE3 = 002340 FILE3A = 002350 FILE3B = 003370 FILE4 = 002414
FILE4A = 002744 FILE5 = 003024 FILE6 = 003030 FILE7 = 003170
FILLA = 002710 FILL1T = 002712 FILL1A = 003734 FLE12A = 003474
FLE12B = 003514 FLE12C = 003522 FLE12D = 003600 FRME0 = 005560
FRME1 = 007148 FRME10 = 010504 FRME11 = 010544 FRME12 = 010640
FRME13 = 011312 FRME14 = 011566 FRME16 = 011714 FRME17 = 012400
FRME2 = 012230 FRME3 = 007334 FRME3A = 007724 FRME5 = 010174
FRME6 = 010420 FRME6A = 010462 FRM10 = 010604 FRM14A = 011572
FRM14B = 011574 FRM16A = 012302 FRM16B = 013370 GRAFH = 120000
GRAPHY = 124000 GREEN = 000002 GSAC0 = 001000 GSRL = 001004
GSVCT = 001002 HERE = 004330 HIGHWR = 001302 HITCNT = 005050
HOLD = 001042 ICNT = 001006 INCR = 001000 INTX = 040000
INT0 = 002000 INT1 = 002200 INT2 = 002400 INT3 = 002600
INT4 = 003000 INT5 = 003200 INT6 = 003400 INT7 = 003600
ITAL0 = 000048 ITAL1 = 000060 KBOCT = 003332 KBOCT = 012465
KRBD = 002040 KYT1 = 002014 KYT2 = 001774 KYT3 = 002024
KYT4 = 002036 KYT5 = 001736 LADVT = 003714 LINE0 = 000004
LINE1 = 000009 LINE2 = 000006 LINE3 = 000007 LOADAC = 004534
LOADBF = 002024 LOAOSP = 002564 LOADUP = 001556 LOADVT = 003704
LOGICA = 004316 LOKRB = 001052 LONGV = 110000 LODPA = 003062
LOCPA1 = 003070 LOOPA2 = 003114 LOOPA3 = 003140 LOOPB = 003214
LOCPB1 = 003222 LOOPB2 = 003246 LOOPB3 = 003272 LOWWR = 001250
LOK3V = 001300 LPDARK = 000300 LPLITE = 000200 LPOFF = 000100
LPON = 000140 LPPNT = 011262 LPVCT = 001070 LPVCT1 = 001072
MAX3X = 017600 MAXSY = 000077 MAXX = 001777 MEXGAA = 000444
MESG = 005412 MESGA = 005430 MINSUY = 000100 MESGAB = 005514
MESGB = 005464 MESGBA = 005514 MINU3X = 020000
MINUSY = 000000 PC = X000007 PNT16A = 012240 PNT16B = 012244
PNT16C = 012254 PNT16D = 012254 PNT16E = 001010
RAY14B = 012006 R = 000777 RAYLPA = 011250 RAYLTV = 010000
RETL = 013550 REQ = 000003 RELATV = 130000 RETB = 001700
RET21 = 012230 RET17 = 005206 RET14 = 000014 RET20 = 002240
R3 = X010003 R4 = X000000 R5 = X000000 R6 = X000000
SCOPEA = 011100 SCOPEB = 001156 SCOPEC = 001162 SCOPEE = 001136
SCOPEF = 001136 SETUPB = 001664 SETUPC = 001664
SETUPA = 011022 SP = X000006 SIZEL = 000000 START = 001354

```

SYTSA = 17000*	SYTSB = 174000	SYCHAR 000000	SYKPR = 000000
SYRA 001400	SYRB 001424	SYRC 001450	SWITCH 000000
SYDOFF = 000010	SYNDN = 000014	SYN12 001050	TAB16A 012010
TAE16B 012130	TIMEVT 001074	TKB 001014	TKS 001010
TIMEVT1 001076	TSAVE 001044	XPOS 001000	YPOS 001000
11 = 012472			

ERRORS DETECTED: 0